5

10 METHOD AND APPARATUS FOR GENERATING
A CROSS PLOT IN ATTRIBUTE SPACE FROM A
PLURALITY OF ATTRIBUTE DATA SETS AND
GENERATING A CLASS DATA SET
FROM THE CROSS PLOT

15

BACKGROUND OF THE INVENTION

This invention relates to the field of seismic data interpretation and,
20 more particularly, to the processing of databases of well log and
geophone-obtained seismic information to better assist interpreters in
classifying subsurface formations.

The subject matter of the present invention relates to a method and
25 apparatus for generating a cross plot in attribute space in response to a
plurality of attribute data and then generating a classified result or "class
data set" based on the data alignment in the crossplot. More
particularly, the subject matter of the present invention relates to an
apparatus and associated method including a computer workstation
30 having a novel software package stored therein, the workstation having a
display screen for displaying a novel classified result, known as a class
data set, for viewing by a workstation operator. The class data set is
generated and displayed on the display screen when the workstation
processor executes the novel software package. When the novel software
35 package is executed, a crossplot in attribute space is generated in
response to a plurality of different attribute data obtained in response to
a seismic operation, and the class data set is generated in response to
the crossplot.

1

In the oil industry, well log data (obtained from well logging operations in wellbores) and/or seismic data (obtained from seismic operations) are obtained from underground earth formations, the data inherently

5   containing information relating to the quantity or degree of underground deposits of hydrocarbons in the earth formations. Computer workstations in conjunction with software stored therein are used to process and interpret the data. The workstations include a processor, a memory for storing interpretation software, and an interactive display

10   screen. The workstation will receive the well log or seismic data and, when the interpretation software stored in the workstation memory is executed by the processor, the workstation will interpret the data and display a novel set of results on the display screen. The set of results will include a novel visual display for viewing by an operator. The display

15   may, for example, reveal the degree or quantity or type of underground deposits of hydrocarbons (e.g., oil) in the earth formation. One such novel set of results displayed on the workstation display screen is known as a "crossplot" and a "class data set". A class data set represents a top view of a subsurface in an earth formation, the top view of the

20   subsurface including, for example, a plurality of different earth formation classes where each class of the plurality of classes in the subsurface possesses a different set of characteristics. When the operator of the workstation views the class data set on the workstation display screen, the characteristics of each class of the subsurface being displayed can be

25   determined.

In the seismic interpretation field, there is a strong need for such a class data set, since the class data set would assist the workstation operator to determine, for example, the type of and/or the existence of underground

30   deposits of hydrocarbon in the earth formation.

2

SUMMARY OF THE INVENTION

Accordingly, it is a primary object of the present invention to improve

5   upon the interpretation systems of the prior art by providing a new interpretation system hereinafter called a "seismic classification system" which is adapted for generating a classified result that can be recorded and displayed on a workstation display.

10   It is a primary aspect of the present invention to provide a new seismic classification system which will produce a classified result by: (a) receiving at least a first attribute data set where the attribute data set includes a plurality of points and a plurality of attribute data corresponding, respectively, to the plurality of points, (b) generating a

15   first cross plot in attribute space in response to the attribute data on the first attribute data set, and (c) generating the classified result in response to the plurality of points on the first attribute data set and the cross plot in attribute space.

20   It is a further aspect of the present invention to provide the new seismic classification system which will produce the classified result, wherein the receiving step (a) for receiving at least a first attribute data set includes the following steps: (a1) reflecting sound vibrations or velocities off the plurality of points on a subsurface, the first attribute data set

25   corresponding to the subsurface, and (a2) generating the plurality of attribute data "(a, b)" associated, respectively, with the plurality of points on the first attribute data set, where "a" could represent, for example, an amplitude value, and "b" could represent, for example, a frequency value.

30   It is a further aspect of the present invention to provide the new seismic classification system which will produce the classified result, wherein the generating step (b) for generating the cross plot in attribute space in response to the attribute data on the first attribute data set includes the

3

following steps: (b1) selecting a subset of the attribute data on the first attribute data set to represent a first set of inspection data, (b2) generating a second cross plot in attribute space from the first set of inspection data, the second cross plot having a distribution/separation

5   of clusters, and (b3) evaluating the distribution/separation of the clusters on the second cross plot.

It is a further aspect of the present invention to provide the new seismic classification system which will produce the classified result, wherein the

10   generating step (b) for generating the cross plot in attribute space in response to the attribute data on the first attribute data set includes the following additional steps: (b4) selecting a subset of the points among a set of clusters on the cross plot in attribute space to represent a second set of inspection data, (b5) generating a second attribute data set from

15   the second set of inspection data, the second attribute data set having a distribution of points, and (b6) evaluating the distribution of the points on the second attribute data set.

It is a further aspect of the present invention to provide the new seismic

20   classification system which will produce the classified result, wherein the generating step (b) for generating the cross plot in attribute space in response to the attribute data on the first attribute data set includes the following additional steps: (b7) selecting some of the attribute data on the first attribute data set to represent training data, (b8) selecting other attribute data on the

25   first attribute data set to represent validation data, a class of the training data being approximately the same as a class of the validation data, (b9) generating a third cross plot in a particular attribute space in response to the training data, and (b10) generating a fourth cross plot in said particular attribute space in response to the validation data, whereby a cluster in said particular attribute

30   space resultant from said validation data should be located adjacent to another cluster in the same said particular attribute space resultant from said training data when said class of said training data is approximately the same as said class of said validation data.

4

It is a further aspect of the present invention to provide the new seismic classification system which will produce the classified result, wherein the generating step (b) for generating the cross plot in attribute space in response to the attribute data on the first attribute data set includes the following additional step and sub-steps: (b11) generating said cross plot in attribute space having at least two clusters of points in response to the attribute data on the first attribute data set when the following three sub-steps have been implemented successfully: (1) the distribution/separation of the clusters on the second cross plot evaluated during the step (b3) is acceptable, (2) the distribution of the points on the second attribute data set evaluated during the step (b6) is acceptable, and (3) said cluster in said particular attribute space resultant from said validation data is located adjacent to said another cluster in the same said particular attribute space resultant from said training data when said fourth cross plot in said particular attribute space is generated during the step (b10).

It is a further aspect of the present invention to provide the new seismic classification system which will produce the classified result, wherein the generating step (c) for generating the classified result in response to the plurality of points on the first attribute data set and the cross plot in attribute space, having said at least two clusters of points, includes the following steps: (c1) assigning a first label, such as color, to a first one of said at least two clusters of points on said cross plot in attribute space, each point in said first one of said at least two clusters of points on said cross plot in attribute space having assigned thereto said first label, such as color, and corresponding to a first set of locations on said first attribute data set, (c2) assigning a second label, such as color, to a second one of said at least two clusters of points on said cross plot in attribute space, each point in said second one of said at least two clusters of points on said cross plot in attribute space having assigned thereto said second label, such as color, and corresponding to a second set of locations on said first attribute data set, (c3) assigning each of said first set of locations on said first attribute data set with said first label, such as color, and

5

(c4) assigning each of said second set of locations on said first attribute data set with said second label, such as color.

In accordance with the above referenced object and the above aspects of the
5   present invention, a novel seismic classification system in accordance with the present invention includes, in combination, a workstation and a novel seismic classification software, known as "Seisclass", stored in the workstation, for performing a novel seismic classification function when the Seisclass software is executed by a processor of the workstation.   The novel seismic classification
10   system of the present invention includes the following system components: (1) "Attribute data sets" which collectively represent the "input attribute data" that are provided to the seismic classification system, the attribute data sets each being shown in a geographical data set coordinate system, the attribute data set being a window for offering a view into the attribute data in accordance with
15   their individual geographic positioning, (2) "Attribute space" which represents a coordinate system having one coordinate axis for each attribute of the aforementioned attribute data, a "crossplot" representing a view into the attribute space, (3) "Administration" which represents the methods for managing the contents of the "input attribute data", for managing the selection
20   of the proper Classification Method, and for managing the parameter control, and (4) a "Class data set" which represents the results of a classification process.

The novel seismic classification system of the present invention further includes
25   the following Classification Methods: (1) "Unsupervised Classification Methods" which look at data clustering/grouping in attribute space, will reveal and confirm natural clustering unbiased by subjective measures but do not necessarily provide a geologically interpretable result, and (2) "Supervised Classification Methods" which adds an "a-priori" knowledge to the classification
30   process as represented by certain selected "training data", the Supervised Classification Methods providing class definitions that can be targeted directly to geology or to the specific problem, and allowing for testable validation by holding out data which have an "a-priori" known class association.

6

However, the Unsupervised classification method and the Supervised classification method each include a different functional operation and therefore enforce a different workflow.

5   For example, the workflow of the Unsupervised classification method includes: (1) selecting a plurality of attribute data sets which have the potential to show a separation of data points in attribute space, the attribute data sets being related to one and the same geological feature (e.g., subsurface, layer in an earth formation), (2) running a classification, (3) evaluating the classification

10   via a crossplot in attribute space relative to a data set view in geographic space, and (4) drawing conclusions by interpreting the grouping in attribute space and perhaps starting a supervised classification to steer the outcome to a more interpretable result.

15   In addition, the workflow of the Supervised classification method includes: (1) the selection of attribute data sets having the potential to show separation of data points in attribute space (the same as in the unsupervised classification method), (2) the selection of training and validation data, the training and validation data each having a known class association and the analysis of the

20   training and the validation data taking place prior to classification, (3) the interaction between a crossplot in attribute space and attribute data set displays in geographic space, (4) prior to running the classification, defining which of the potential training and validation data are used for "training", i.e., data used to build the classification function, and which of the potential

25   training and validation data are "validation", i.e., data withheld from the classification process, (5) running the classification thereby producing a classification result, and optionally various quality and confidence measures, and (6) comparing the predicted class association from the classification result with the withheld validation data, i.e. the a-priori known class association,

30   thereby providing a consistency and stability test.

Further applicability of the present invention will become apparent from the detailed description presented hereinafter. It should be understood,

however, that the detailed description and the specific examples, while representing a preferred embodiment of the present invention, are given by way of illustration only, since various changes and modifications, within the spirit and scope of the invention, will become obvious to one skilled in the art from a reading of the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

A full understanding of the present invention will be obtained from the detailed description of the preferred embodiment presented hereinbelow, and the accompanying drawings, which are given by way of illustration only and are not intended to limit the present invention, and wherein:

figure 1 illustrates a typical seismic operation;

figure 2 illustrates a well logging operation;

figures 3 and 4 illustrate the seismic signals which are received by the geophones of figure 1;

figure 5 illustrates how the seismic data output record and the well log data output record combine to represent the "data received";

figure 6 illustrates how the attribute data sets are generated from the data received of figure 5;

figures 7 through 13 illustrate how the attribute data sets are produced from a seismic operation;

figures 14 and 15 illustrate a workstation which stores the seisclass software of the present invention;

figure 16 illustrates a basic flowchart representing the overall functional operation of the present invention when the seisclass software of the present invention is executed;

5   figures 17 through 21 illustrate how the class data set of the present invention is produced when the attribute data sets of figure 13 are used by the workstation processor of figure 14 to produce the class data set of the present invention during execution of the seisclass software of the present invention;

10

figure 22 illustrates a more detailed construction of the seisclass software of the present invention including an Unsupervised seisclass software and a Supervised seisclass software;

15   figure 23 illustrates a block diagram depicting the Unsupervised seisclass software of the present invention of figure 22;

figure 24 illustrates the principles behind the "auxiliary results QC measures" code 142 of figure 23;

20

figure 25 illustrates a block diagram depicting the unsupervised classification code 132 of figure 23;

figure 26 illustrates a block diagram and description depicting the apply

25   classification and determine quality indicators code 146 of figure 25;

figure 27 illustrates the functional operation of blocks 158, 160, and 162 of figure 26;

30   figure 28 illustrates the functional operation of blocks 166, 168, and 170 of figure 26;

figure 29 illustrates the functional operation of block 172 of figure 26;

9

figure 30 illustrates illustrates a block diagram depicting the Supervised seisclass software of the present invention of figure 22;

5   figure 31 illustrates the various classes which may be selected for the 'classes in use - manually defined class types' of figure 30;

figures 32 through 34 illustrate how one could select and declare training data in connection with the 'training/validation' of figure 30;

10

figures 35 through 37 illustrate how one could select and declare validation data in connection with the 'training/validation' of figure 30;

figure 38 illustrates a detailed construction of the supervised
15   classification code 186 of figure 30;

figure 39 illustrates a detailed construction of the "training of the classifications" code 208 of figure 38;

20   figure 40 illustrates the functional operation of blocks 226, 228, and 230 of figure 39;

figure 41 illustrates the functional operation of block 234, 236, and 238 of figure 39;

25

figure 42 illustrates the functional operation of the "training of the classifications" block 242 of figure 39;

figure 43 illustrates the functional operation of the "apply the trained
30   classification and determine quality indication" 216 of figure 38;

figure 44 illustrates the functional operation of the "classified result" or "class data set" block 130 of figure 30; and

figure 44A illustrates a 3D cube which includes a plurality of such 'class data sets' corresponding, respectively, to a plurality of subsurfaces in an earth formation.

5

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to figure 1, a seismic operation is illustrated. An acoustic source 10 produces sound vibrations 12 which are reflected off a

10 plurality of subsurfaces 14 of an earth formation, the subsurfaces being separated by a fault 16. The use of sound vibrations generated by the acoustic energy source is given by way of example only, since the sound vibrations generated by the acoustic energy source is only a subset of a larger set of all measureable quantities derived from or associated with

15 seismic data, such as velocities which is a property closely related to seismic data. The sound vibrations 12 are received in a plurality of geophone receivers 18, and electrical signals are generated from the receivers, those electrical signals representing the "received seismic data" 20 in figure 1. The received seismic data 20 are provided as input data

20 to a recording truck 22 which includes a recording truck computer 22a. The recording truck computer 22a will receive the "received seismic data" 20 and it will thereafter produce a "seismic data output record" 24 which stores the received seismic data 20.

25 Referring to figure 2, although not shown in figure 1, a wellbore has been drilled in the earth formation of figure 1, that wellbore being illustrated in figure 2. In figure 2, the wellbore 26 penetrates the formation, and a logging tool 28 logs the formation. A logging truck 30 is disposed at the earth surface, the logging truck 30 including a computer 32a which

30 produces a "well log data output record" 32 which records and/or stores information representing the characteristics of the earth formation that is located nearest the wall 26a of the wellbore 26.

Referring to figure 3, an example of the "received seismic data" 20 of figure 1 which is stored on the seismic data output record 24 of figure 1 is illustrated. Note, in figure 3, that the received seismic data 20 is actually comprised of a multitude of seismic "traces", where each seismic

5　trace includes a plurality of variations in amplitude plotted over time. Figure 4 illustrates an exploded view of a portion of a single such seismic trace of figure 3.

Referring to figure 4, an exploded view of a portion of four seismic traces

10　from figure 3 is illustrated. In figure 4, each of the seismic traces 34, 36, 38, and 40 include an amplitude variation 34a, 36a, 38a, and 40a. The amplitude variations 34a through 40a actually represent the location (that is, the depth) in the earth formation where an earth subsurface 14 is located. Figure 4 illustrates the earth subsurface 14 by using a dotted

15　line 14 passing through the four amplitude variations 34a through 40a. Referring back to figure 3, a multitude of such amplitude variations are now visible in the seismic traces representing a plurality of such earth subsurfaces 14 located along a depth dimension of the earth formation.

20　Referring to figure 5, the seismic data output record 24 and the well log data output record 32 combine to produce the "data received" 42 which is input to the mainframe computer of figure 6.

Referring to figure 6, the data received 42 is received by a workstation

25　computer 44. The processor 44a will receive the data received 42, it will execute a "data reduction and attribute generation software" stored in the workstation computer memory 44b, and responsive thereto, it will produce a plurality of "attribute data sets" 46. One such "data reduction software" is disclosed in a book called "Seismic Velocity Analysis and the

30　Convolutional Model", by Enders A Robinson, the disclosure of which is incorporated by reference into this specification. An attribute data set 46 includes a first "attribute data set" consisting of a plurality of parameters of a first type and at least a second "attribute data set" consisting of a

plurality of parameters of a second type. For example, one attribute data set 46 could include a first "attribute data set" consisting of a set of amplitude parameters, and a second "attribute data set" consisting of a set of frequency parameters.

Referring to figures 7 through 13, a simple example is illustrated which will demonstrate how the attribute data sets 46 of figure 6 are generated by the workstation computer 44.

In figure 7, an acoustic source 50 (which is given by way of example only since all measurable quantities associated with seismic data could be utilized) produces sound vibrations 52 which reflect off a subsurface layer 54 in an earth formation. The sound vibrations 52 are reflected off the subsurface layer 54 and are received in a plurality of geophones 56 disposed on the earth's surface. Electrical signals 58 are input to a computer 60a of a recording truck 60. In figure 7, the subsurface layer 54 includes at least eight (8) points or cells: point A, point B, point C, point D, point E, point F, point G, and point H. The subsurface layer 54 also has an x-y coordinate system imposed thereon, where the x-axis along the subsurface layer 54 includes coordinate values "x1" through "x8", and the y-axis along the subsurface layer 54 includes coordinate values "y1" and "y2". From figure 7, it is evident that points A through H on subsurface layer 54 have the following (x, y) coordinate values which identify their physical geographic locations on the subsurface 54:

A (x1, y1)   E (x5, y1)
B (x2, y2)   F (x6, y2)
C (x3, y1)   G (x7, y1)
D (x4, y2)   H (x8, y2)

In figure 7, an oil well 60 is located above a wellbore 62 which penetrates an earth formation. The wellbore 62 intersects with and penetrates the subsurface layer 54 in the earth formation of figure 7. In addition, it

13

appears from figure 7 that the reflections of the sound vibrations 52 off the subsurface layer 54 are non-vertical. However, this is not quite true. Figures 8 and 9 will correct any misconceptions which are generated by the figure 7 illustration. Before referring to figure 8, however, refer to

5 figure 7 again and note that the sound vibrations 52 reflecting off the subsurface layer 54 actually include the following individual sound vibration reflections: a first sound vibration reflection 52a reflecting from point A on subsurface layer 54, a second sound vibration reflection 52b from point B, a third sound vibration reflection 52c from point C, a

10 fourth sound vibration reflection 52d from point D, a fifth sound vibration reflection 52e from point E, a sixth sound vibration reflection 52f from point F, a seventh sound vibration reflection 52g from point G, and an eighth sound vibration reflection 52h reflecting from point H on subsurface layer 54.

15

In figure 8, the sound vibration reflections 52a through 52h actually reflect substantially vertically off the points A through H on the subsurface layer 54 when such sound vibration reflections propagate between points A through H on the subsurface layer 54 and the

20 geophone receivers 56 on the earth's surface. However, as noted in figure 9, when such sound vibrations reflect upwardly from any one particular point (A through H) on the subsurface layer 54, there exist several such reflections.

25 In figure 9, using point D on subsurface layer 54 as an example, a sound vibration reflection 52d reflects upwardly off the point D on the subsurface layer 54 and it is received in a geophone receiver 56a; however, other sound vibration reflections 52d1 through 52d8 also reflect off the point D and are received in other geophone receivers 56b

30 through 56i which are also positioned on the earth's surface.

Refer back to figures 3 and 4, and recall that figure 3 comprises a multitude of seismic traces, similar to the seismic traces 34, 36, 38, and 40 in figure 4. In addition, recall that each of the seismic traces in figure

14

3 include a plurality of amplitude variations similar to the amplitude variations 34a, 36a, 38a, and 40a in the seismic traces 34 through 40 in figure 4.

5   In figure 10, a method for determining at least one (1) pair of "attributes" associated with each point (e.g., points A through H) on the earth's subsurface layer 54, off which one of the sound vibrations 52 of figure 7 are reflected, is illustrated.

10   In figure 10, a seismic trace having an amplitude variation is associated with each of points A, B, C, and D on the subsurface layer 54. Although not shown in figure 10, a seismic trace having an amplitude variation is also associated with points E, F, G, and H on the subsurface layer 54. In figure 10, locate point A on the subsurface layer 54, and notice that

15   point A has a seismic trace 64 which includes an amplitude variation 64a, the amplitude variation 64a having an amplitude "a6" and a frequency "f2". Point B has a seismic trace 66 which includes an amplitude variation 66a, the amplitude variation 66a having an amplitude "a3" and a frequency "f5". Point C has a seismic trace 68

20   which includes an amplitude variation 68a, the amplitude variation 68a having an amplitude "a5" and a frequency "f3". Point D has a seismic trace 70 which includes an amplitude variation 70a, the amplitude variation 70a having an amplitude "a4" and a frequency "f6". Therefore, in figure 10, for point A on the subsurface layer 54, one "attribute" 74

25   associated with point A is amplitude "a6" 74 and another "attribute" 76 associated with point A is frequency "f2" 76. Similarly, for point B, one "attribute" 78 associated with point B is amplitude "a3" 78 and another "attribute" 80 associated with point B is frequency "f5" 80. For point C, one "attribute" 82 associated with point C is amplitude "a5" 82 and

30   another "attribute" 84 associated with point C is frequency "f3" 84. For point D, one "attribute" 86 associated with point D is amplitude "a4" 86 and another "attribute" 88 associated with point D is frequency "f6" 88.

However, in figure 10, a seismic trace having an amplitude variation is also associated with the wellbore 62. That is, in figure 10, a seismic trace 72 having an amplitude variation 72a is also associated with the wellbore 62. The amplitude variation 72a has an amplitude "a2" and a

5 frequency "f4". Therefore, for the wellbore 62, one "attribute" 90 associated with the wellbore 62 is amplitude "a2" 90 and another "attribute" 92 associated with the wellbore 62 is frequency "f4" 92. Actually, the attributes 90, 92 associated with the wellbore 62 are "synthesized". That is, the wellbore 62 in figure 10 will produce wellbore

10 data similar to the well log output record 32 of figure 2; and, from that wellbore data, the seismic trace 72 in figure 10 is "synthesized"; and, from that seismic trace 72, the attribute 90 (amplitude "a2") and the attribute 92 (frequency "f4") are generated.

15 In figure 11, recall from figure 7 that points A through H on subsurface layer 54 have the following (x, y) coordinate values:

A (x1, y1)   E(x5, y1)

B (x2, y2)   F(x6, y2)

20 C (x3, y1)   G(x7, y1)

D (x4, y2)   H(x8, y2)

In figure 12, the subsurface layer 54 is again illustrated, the subsurface layer 54 including our example points A, B, C, D, E, F, G, and H.

25

Using the "method for determining at least one (1) pair of 'attributes' associated with each point (A through H) on the earth's subsurface layer 54" that was discussed above with reference to figure 10, let us now assume, for purposes of our example, that the "attributes" for point A on

30 subsurface layer 54 are (f2, a6), where "f2" is a frequency and "a6" is an amplitude. Similarly, assume that the "attributes" for point B on subsurface layer 54 are (f5, a3).

Refer now to figure 12, and assume, for purposes of our example, that points A through H on the subsurface layer 54 in figure 12 have the following (x, y) "coordinates" and points A through H also have the following (f, a) "attributes", where "f" designates frequency and "a" designates amplitude:

The point on subsurface layer 54	(x, y) coordinates	(f, a) "attributes"
A	(x1, y1)	(f2, a6)
B	(x2, y2)	(f5, a3)
C	(x3, y1)	(f3, a5)
D	(x4, y2)	(f6, a4)
E	(x5, y1)	(f1, a3)
F	(x6, y2)	(f5, a4)
G	(x7, y1)	(f4, a7)
H	(x8, y2)	(f3, a2)

In figure 13, by using the (x, y) coordinates and the (f, a) "attributes" derived above with reference to figure 12, we can now determine the "Attribute Data Sets" 46 which were generated by the computer 44 of figure 6. In figure 13, one attribute data set 54a of the "attribute data sets" 46 of figure 6 would comprise the amplitude "attribute", and another attribute data set 54b of the "attribute data sets" 46 of figure 6 would comprise the frequency "attribute".

In figure 13, therefore, the one attribute data set 54a of the "attribute data sets" 46 comprised of the amplitude "attribute" would include the following data:

The point on subsurface 54	(x, y) coordinates --	amplitude "attribute"
A	(x1, y1)	(a6)
B	(x2, y2)	(a3)
C	(x3, y1)	(a5)
D	(x4, y2)	(a4)
E	(x5, y1)	(a3)
F	(x6, y2)	(a4)
G	(x7, y1)	(a7)
H	(x8, y2)	(a2)

In figure 13, another attribute data set 54b of the "attribute data sets" 46 comprised of the frequency "attribute" would include the following data:

5

The point on subsurface 54	(x, y) coordinates	frequency "attribute"
A	(x1, y1)	(f2)
B	(x2, y2)	(f5)
C	(x3, y1)	(f3)
D	(x4, y2)	(f6)
E	(x5, y1)	(f1)
F	(x6, y2)	(f5)
G	(x7, y1)	(f4)
H	(x8, y2)	(f3)

Referring to figures 14 and 15, a computer workstation 100 is illustrated.

In figure 14, the workstation 100 includes a workstation processor 100a
10   and a workstation memory 110b, the workstation processor 100a being
connected to a system bus, and the workstation memory 100b being
connected to the system bus. The Attribute Data Sets 46, generated by
the computer 44 of figure 6 and illustrated in figure 13, are provided as
"input data" to the workstation 100, and a workstation "Display and

18

Interaction" 100c is visible to a workstation operator, the display 100c including one or more windows which are used by the operator for "interaction" purposes to manipulate subsequent operations. The workstation memory 100b stores a novel software package in accordance

5   with the present invention, hereinafter known as the "Seisclass Software". When the Seisclass software 100b is executed by the workstation processor 100a, a variety of window displays are presented to the operator on the Display 100c, the window displays being used by the operator to execute and perform certain "interaction" functional

10   operations which are designed for manipulating the functional operation of the seisclass software (those manipulations will be discussed later in this specification).

In figure 15, the workstation 100 includes a monitor/display 100c for
15   performing the "interaction", a processor 100a which includes the memory 100b for storing the "seisclass software", a keyboard 100d, and a mouse 100e. A storage medium 102, usually a "CD-Rom" 102, initially stores the "seisclass software" 100b of the present invention. The CD-Rom 102 is inserted into the processor 100a of the workstation
20   100 in figure 15, and the "seisclass software", stored on the CD-Rom 102, is read from the CD-Rom 102 and is loaded into the workstation memory 100b. The monitor/display 100c in figure 15 is a cathode ray tube which is adapted for displaying a plurality of "window displays" 104 thereon that are used for "interaction" purposes. The word "interaction"
25   implies an operator/window display interaction, whereby the operator would use the mouse 100e to place a cursor in a window display 104, click on the mouse 100e, and thereby perform the "interaction" involving the execution of a portion of the seisclass software 100b.

30   Referring to figure 16, a flowchart 106 is presented which illustrates the functional operation performed by the seisclass software 100b when the seisclass software 100b is executed by the workstation processor 100a of figure 14. In figure 16, the first two blocks 108 and 110 of the flowchart

106 have already been discussed: block 108 "Reflect Seismic Energy off of a Plane in the Earth Formation subdivided into grids in an x-y coordinate system" (figures 7 through 10), and block 110 "Collect a Plurality of Attribute Data sets from Seismic Reflections" (figures 11 through 13).

However, block 112 of the flowchart 106 in figure 16 illustrates the function performed by the workstation processor 100a when the "seisclass software" 100b of the present invention is executed by the workstation processor 100a of figure 14. Block 112 of flowchart 106 comprises the following steps:

(1) "Produce a cross plot in attribute space from the attribute data sets", sub-block 112a,

(2) "Subdivide the cross plot into a plurality of zones comprised of points which correspond to other points on the attribute data sets where each zone on the cross plot has a different class association than any other zone which is indicated by a different label, such as color", sub-block 112b, and

(3) "Produce a Class Data Set comprised of a plurality of points on the attribute data sets where each point has a label, such as color, depending on its cluster/class association on the cross plot in attribute space", sub-block 112c.

The "Class Data Set", otherwise called a "classified result", is displayed on the workstation "Display" 100c of figure 14 and 15.

Referring to figure 17 through 21, when the Seisclass software of the present invention (represented by block 112 of the flowchart 106 of figure 16) is executed by the workstation processor 100a of figure 14, a functional operation is performed by that workstation processor 100a.

That functional operation is set forth in the following paragraphs with reference to figures 17 through 21.

In figure 17, recall from the above discussion with reference to figure 12 and 13 that the points A through H on the subsurface layer 54 have a set of (x, y) coordinates and a corresponding set of (f, a) "attributes". That set of (x, y) coordinates and that set of (f, a) "attributes" are duplicated again in figure 17.

In figure 18, let us now plot the (f, a) coordinates in an "amplitude" - "frequency" coordinate system, where "amplitude" is plotted on the y-axis and "frequency" is plotted on the x-axis. When the (f, a) "attribute" coordinates of figure 17 are plotted in an "amplitude" - "frequency" coordinate system, the result of that plot is shown in figure 18 as element numeral 114. The plot shown in figure 18 is hereinafter called a "cross plot in attribute space" or "cross plot 114". The (f, a) "attribute" coordinates of figure 17 were carefully selected to demonstrate the "clustering" now visible in the cross plot in attribute space illustrated in figure 18. In the cross plot 114 of figure 18, note that points A, G, E, and C are "clustered" together in a "cluster" 116, and points F, D, H, and B are separately "clustered" together in a "cluster" 118. Let us now assign a label, such as color, to the cluster 116, and assign a separate and different label, such as color, to the cluster 118. The use of color as a label in figure 18 is given by way of example only, since other kinds of labels other than color, such as numeric tags, might be more meaningful. The color "red" is assigned to the cluster 116 in the cross plot 114 of figure 18, and the color "green" is assigned to the cluster 118 in the cross plot 114. Therefore, the points A, G, E, and C have been assigned the color "red", and the points F, D, H, and B have been assigned the color "green".

In figure 19, an actual cross plot 114 is illustrated. Note the clusters 120, 122, 124, and 126 in attribute space.

From the above discussion, it is apparent that the points A through H on
the subsurface layer 54 in the earth formation have the following
(x, y) coordinates and the following labels (i.e.,"colors" in our example):

TABLE 1

Point	(x, y) coordinates	Color label
A	(x1, y1)	red
B	(x2, y2)	green
C	(x3, y1)	red
D	(x4, y2)	green
E	(x5, y1)	red
F	(x6, y2)	green
G	(x7, y1)	red
H	(x8, y2)	green

In figure 20, let us now plot the points A through H on subsurface layer
54 in an (x, y) coordinate system, using the above referenced (x, y)
coordinates set forth in the above table 1, and then assign to each
plotted point the "color" label indicated in the above table 1. The result
of that plot (hereinafter called a "Class Data Set" 130) is shown in figure
20.

In figure 21, an actual "Class Data set" plot 130 is shown in figure 21
(actually, the actual Class Data Set plot would be illustrated in a
plurality of different colors, but, for purposes of this specification, cross
hatching is utilized). The "Class Data set" plot 130 of figure 21 is shown
and displayed on the "Display" 100c of the workstation 100 of figures 14
and 15. In addition, the "Class Data set" plot 130 of figure 21 can be
recorded on a recorder that is connected to the system bus in of the
workstation 100 of figure 14. The "Class Data set" 130 is also visible to
an operator sitting at the workstation 100, the "Class Data set" being the
end result that is produced in response to the execution of the "seisclass
software" 100b of the present invention.

22

In the "Class Data set" plot of figure 21, different colors are indicated by a different shading (although, the real Class Data set will display colors and not shading). When the frequency and amplitude "attributes" associated with a "plurality of points" on the subsurface layer 54 of figure 7 are approximately the same, the labels, such as colors, associated with those "plurality of points" on the Class Data set 130 of figure 21 will also be the same.

On the other hand, when the frequency and amplitude "attributes" associated with a sound vibration reflecting from "first plurality of points" on the subsurface layer 54 are different from the frequency and amplitude "attributes" associated with a sound vibration reflecting from a "second plurality of points" on the subsurface layer 54 of figure 7, then the label, such as color, associated with that "first plurality of points" on the Class Data set 130 will be different from the label, such as color, associated with that "second plurality of points" on the Class Data set 130.

For example, if the "first plurality of points" reside on an area of the subsurface layer 54 which is "sand", and the "second plurality of points" also reside on an area of the subsurface layer 54 which is also "sand", then the frequency and amplitude "attributes" (as well as other such "attributes") associated with a sound vibration reflecting from the "first plurality of points" on the subsurface layer 54 will be the same as the frequency and amplitude "attributes" associated with a sound vibration reflecting from the "second plurality of points" on the subsurface layer 54 of figure 7. As a result, the color label associated with that "first plurality of points" on the Class Data set 130 will be the same as the color label associated with that "second plurality of points" on the Class Data set 130.

On the other hand, if the "first plurality of points" reside on an area of the subsurface layer 54 which is "sand", and the "second plurality of points" reside on an area of the subsurface layer 54 which is "brine" (i.e., not the same as 'sand'), then the frequency and amplitude

5   "attributes" (as well as other such "attributes") associated with a sound vibration reflecting from the "first plurality of points" on the subsurface layer 54 will be different from the frequency and amplitude "attributes" associated with a sound vibration reflecting from the "second plurality of points" on the subsurface layer 54 of figure 7. As a result, the color label

10   associated with that "first plurality of points" on the Class Data set 130 will be different from the color label associated with that "second plurality of points" on the Class Data set 130. Therefore, the different color labels visible on the Class Data set between a "first set of points" and a "second set of points" indicate that the set of characteristics (i.e.,

15   the amplitude, frequency, reflection intensity, etc) associated with the reflections of the sound vibrations from the "first set of points" on the subsurface layer 54 of figure 7 is different than the set of characteristics associated with the reflections of the sound vibrations from the "second set of points" on the subsurface layer 54 of figure 7.

20

When all the "set of characteristics" associated with all the points on the subsurface layer 54 of figure 7 are known, the operator sitting at workstation 100 of figures 14 and 15 can study the resultant "Class Data set" 130 and, from that "Class Data set", the operator can more

25   accurately determine the possibility of the existence of oil or other hydrocarbon bearing formations located adjacent or near that subsurface layer 54.

Referring to figure 22, the actual structure of the "seisclass software"

30   100b of figure 14 will be set forth in the following paragraphs with initial reference to figure 23. Before starting discussion of the structure of the "seisclass software" 100b, in figure 22, it should be noted that there are

24

two types of "seisclass software" 100b: (1) an "Un-supervised" seisclass software 100b1, and (2) a "Supervised" seisclass software 100b2.

The "Un-supervised" seisclass software 100b1 is used when the "classes" of the data present within the "attribute data sets" 46 of figure 14 are not known, and the "supervised" seisclass software 100b2 is used when the "classes" of the data present within the "attribute data sets" 46 are known. The word "classes" means the class or category to which the data in the attribute data sets 46 belong. For example, the data may belong to one of the following classes: sand or shale or gas or fluid or oil or brine, etc. The classes will be discussed later in this specification.

Referring to figures 23 through 29, the structure of the "Un-supervised" seisclass software 100b1 of figure 22 is illustrated.

In figures 23 and 10, referring initially to figure 23, the Un-supervised seisclass software 100b1 comprises an "unsupervised classification" block of code 132 which is responsive to and receives three sets of input data: (1) input attributes 134, (2) well supplement attributes 136, and (3) parameters controlling the classification process 138. The input attributes 134 and the well supplement attributes 136 can best be understood by referring back to our simple example shown in figure 10. In figure 10, the input attributes 134 are the following: (1) the amplitude attribute "a6" 74 and the frequency attribute "f2" 76 for point A, (2) the amplitude attribute "a3" 78 and the frequency attribute "f5" 80 for point B, (3) the amplitude attribute "a5" 82 and the frequency attribute "f3" 84 for point C, and (4) the amplitude attribute "a4" 86 and the frequency attribute "f6" 88 for point D. On the other hand, in figure 10, the well supplement attributes 136 are the following: the amplitude attribute "a2" 90 and the frequency attribute "f4" 92 for the wellbore 62. Recall, from figure 10, that the wellbore seismic trace 72 is synthesized from the well log data obtained when the wellbore 62 is logged by a logging tool, as depicted in figure 2. Referring back to figure 23, the "parameters

25

controlling the classification process" 138 refers to the parameters associated with the classification method in the "unsupervised classification" 132. Briefly, the seisclass software 100b offers a selection of classification methods, and each such method has its own particular "parameters" which are closely related to the particular method. All parameters that are required by a classification method are defined in a "method/parameter catalogue". As noted below, when the "parameters" are selected, those "parameters" can control the classification process performed by the "unsupervised classification" 132. In addition, editing the "parameters" 138 will modify slightly the classification process performed by the "unsupervised classification" 132.

In figure 23, in response to the input attributes 134 and the well supplement attributes 136 and the parameters controlling the classification process 138, the unsupervised classification code 132 will produce a "classified result" 130, the "classified result" 130 being the aforementioned "Class Data set" 130. That is, in response to the input attributes 134 and the well supplement attributes 136 and the parameters controlling the classification process 138, the unsupervised classification code 132 will generate a cross plot in attribute space (similar to the cross plot 114 of figure 18) and then, responsive to the cross plot, the unsupervised classification code 132 will generate a "classified result" which is the Class Data set 130 (similar to the Class Data set 130 shown in figures 20 and 21).

In figure 23, when the classified result 130 (the Class Data set 130) is produced, a quality control (QC) measure is implemented. After the class data set 130/classified result 130 is produced, the "Auxiliary Results/QC Measures" code 142 in figure 22 will help to assess the quality of the classified result 130, that is, to determine the degree of accuracy of the information set forth in the Class Data set 130 or classified result 140. Refer now to figure 24 to understand how the "Auxiliary Results/QC Measures" code 142 performs this function.

In figure 24, the cross plot 114 of figure 18 is illustrated again, showing the two clusters of data points 116 and 118. Note, however, that a point "P1" is associated with cluster 116 and point "P2" is associated with

5   cluster 118, yet, points P1 and P2 are outside their respective clusters. That is, point P1 is outside its cluster 116 by a distance "y1" from the center of the cluster 116, and point P2 is outside its cluster 118 by distance "y2" from the center of the cluster 118.

10   When the "unsupervised classification" code 132 of figure 22 plots the cross plot in attribute space 114 in response to all of the input attributes 134 and the well supplement attributes 136 and the parameters controlling the classification process 138, the "Auxiliary Results/QC Measures" code 142 will notice that points P1 and P2 are outside their

15   respective clusters 116 and 118 by distances "y1" and "y2". Since the distances "y1" and "y2" are quite large, the "Auxiliary Results/QC Measures" code 142 will assign a "lower confidence level" to the classification of points P1 and P2. As a result, in view of the aforementioned "lower confidence level" with regard to the classification

20   of points P1 and P2, the "Auxiliary Results/QC Measures" code 142 in figure 23 has determined that the degree of accuracy of the information set forth in the classified result 130 (Class Data set 130) is less than previously expected.

25   In figure 25, the structure and functional operation of the "unsupervised classification" code 132 of figure 23 is illustrated.

In figure 25, the "unsupervised classification" code 132 will receive the "input attribute selection", the input attribute selection including the

30   "input attributes" 134 and the "well supplement attributes" 136. The "unsupervised classification" code 132 will also receive the "parameters controlling the classification process" 138 (the input attribute selection 134, 136 and the parameters controlling the classification process 138

being hereinafter collectively called "input"). Note, in figure 25, that the "parameter set" data block 138 is input to a summing element 139 via a first line 141; however, a second line 143 from an edit block 154 (discussed below) is also input to the summing element 139. The

5 aforementioned "input" will be saved via the "save input" block 144 in figure 24 (i.e., the composition of the input attribute selection 134, 136 and the parameters 138 will be saved via the "save input" block 144 in order to re-run the classification to reproduce the results or to retain the composition as a reference for further tests using different components).

10 In addition, that "input" will be received by the "apply classification and determine quality indicators" code 146 in figure 25 (see figure 26 for a construction of the "apply classification..." code 146). The "apply classification..." code 146 will then use the "parameters..." 138 to "apply the classification". That is, the "apply classification..." code 146 will use

15 the "parameters..." 138 to produce a "cross plot in attribute space". In that cross plot, the "input attributes" 134 and the "well supplement attributes" 136 will be clustered in the same manner as discussed above with reference to figure 18 when the cross plot 114 in attribute space having clusters 116 and 118 were produced. When the "apply

20 classification..." code 146 completes its function of producing the "cross plot in attribute space", the "parameters..." 138 are saved, via the "save parameters" block 148, and the resultant "cross plot in attribute space" is also saved, via the "save results" block 150. Next, the "unsupervised classification" code 132 asks, via the "quality acceptable?" block 152 in

25 figure 24, whether the "cross plot in attribute space", produced by the "apply classification..." code 146, is acceptable from a quality standpoint. If the quality of the "cross plot in attribute space" is not acceptable when the "quality acceptable" code 152 is executed, the unsupervised classification code 132 is directed to the "edit block" 154 in figure 24. In

30 the edit block 154, two methods of "editing" are implemented: (1) "edit input attribute selection" 154a (wherein the "input attribute selection" 134, 136 are edited/changed), and (2) "edit parameters" 154b (wherein the parameters controlling the classification process 138 are

edited/changed). Following the editing performed by the edit block 154, the "apply classification..." code 146 will use the edited/changed input attributes 134 and the edited/changed well supplement attributes 136 and the edited/changed parameters controlling the classification process

5   138 to re-apply the classification (i.e., to produce a "new cross plot in attribute space"). If the quality of the "new cross plot in attribute space" is acceptable (when the "quality acceptable" block 152 is executed), the "new cross plot in attribute space" is passed to the "classified result" 130 block of code in figure 22 where the Class Data set/classified result 130

10   will be plotted in response to the "new cross plot in attribute space".

In figure 26, the structure and functional operation of the "apply · classification and determine quality indicators" code 146 of figure 25 is illustrated.

15

In figure 26, starting with the summing element 139, the input attributes 134, 136 (situated on the "attribute data sets" 54a and 54b of figure 13) and the parameters controlling the classification process 138 are input to the summing element 139 via a first line 141. An operator sitting at

20   workstation 100 of figure 15 has an opportunity, at this point, to inspect the input attributes 134, 136 which are situated on their "attribute data sets" (of figure 13) relative to the "cross plot in attribute space" (of figure 18) which results from those input attributes by viewing the contents of the plurality of window displays 104 on the monitor of the workstation

25   100 of figure 15. If a manual inspection of the input attributes 134, 136 situated on the "attribute data sets" (of figure 13) relative to the resultant "cross plot in attribute space" (of figure 18) is necessary, the operator at the workstation 100 will indicate that such an inspection is necessary, thereby triggering the execution of the first "manual inspection" block

30   156 in figure 26 (note that there are two "manual inspection" blocks in figure 26, a first "manual inspection" block 156, and a second "manual inspection" block 164). Assume that the operator indicated that a manual inspection of the input attributes 134, 136 on their "attribute

data sets" (of figure 13) relative to the resultant "cross plot in attribute space" (of figure 18) is necessary and that, as a result, the "yes" output from the first "manual inspection" block 156 in figure 25 is generated.

5   In figures 26 and 27, if the "yes" output from the first "manual inspection" block 156 of figure 26 is generated, the following portions of the "apply classification..." code 146 in figure 26 will be executed:

(Step 1) "select inspection data" 158 (figure 26),

10   (Step 2) "generate plot in attribute space" 160 (figure 26), and
(Step 3) "evaluate distribution/separation" 162 (figure 26).

In figure 27, inspection data 173 is selected (block 158 in figure 26). The inspection data 173 of figure 27 are comprised of a plurality of input

15   attributes (a_x, b_x) [such as the amplitude and frequency attributes "(f, a)" of figure 12] that are a subset of a larger plurality of input attributes 177 on the attribute data set "a"/ attribute data set "b" of figure 27.

In response to the larger plurality of input attributes 177 on the attribute

20   data sets "a" and "b", a "cross plot in attribute space" 179 is generated in figure 27 and, in our example, four clusters are formed in the "cross plot in attribute space" of figure 27.

However, in figure 27, the inspection data 173 on the attribute data set

25   "a" and the attribute data set "b" in figure 27 [including a plurality of the input attributes (a_x, b_x)] are selected (block 158 in figure 26), and a "cross plot in attribute space" 181a and 181b in figure 27 is generated from that inspection data 173 (block 160 in figure 25). The distribution-separation of that "cross plot in attribute space" 181a, 181b of figure 27

30   is evaluated (block 162 of figure 25).

In figure 26 and 28, if an additional manual inspection of the "cross plot in attribute space" 179 of figure 27 relative to the corresponding distribution

of the (a_x, b_x) input attributes on the "attribute data sets" is required, a "yes" output is generated from the second "manual inspection" block 164 in figure 26. As a result, the following additional portions of the "apply classification..." code 146 in figure 26 will be executed:

(Step 4) "select inspection data" 166 (figure 26),
(Step 5) "generate data set plot" 168 (i.e., the "attribute data set" 54a, 54b), and
(Step 6) "evaluate distribution separation" 170 (on the "attribute data set").

In figure 28, when the "cross plot in attribute space" 179 is generated from the larger plurality of input attributes 177 on the attribute data sets "a" and "b", additional "inspection data" 185 is selected from among some of the clustered points in the "cross plot in attribute space" 179 (block 166 of figure 26). Using that "inspection data" 185 on the "cross plot in attribute space" 179, another "attribute data set" plot 187 is generated from that inspection data 185 (block 168 of figure 26). The distribution-separation of the input attributes (a_x, b_x) on said another "attribute data set" plot 187 in figure 28 is evaluated (block 170 of figure 26).

In figure 26, steps 1 through 3 above involve the clustering of "some" of the input attributes 134, 136 (called "inspection data") in attribute space, similar to the clustering of the attributes illustrated in figure 18, and evaluating the separation of the resultant clusters of the data to determine if such clustering is acceptable. It should be noted here that classification might still be viable even if the cluster separation is poor. The results might be less valuable, but those results might still be useful. Steps 4 through 6 involve the selection of some of the points within a cluster or between clusters in attribute space (further inspection data), the generation of an "attribute data set" plot from that further inspection data, and the evaluation of the distribution of the input attributes on the resultant "attribute data set" plot.

31

The assessment of the resultant "attribute data set" is done in an interactive fashion; that is, the resulting "intermediate attribute data set" is inspected visually and it is compared with its corresponding "cross plot in attribute space". The pointer-position in the attribute data set (i.e., the data set domain) identifies and highlights the respective sample in the crossplot in attribute space. Similarly, points in the cross plot in attribute space can be used to highlight the corresponding points in the "attribute data set". After the inspection, inherent in steps 1 through 6, is complete, the operator may draw different conclusions: (1) return to the input attribute selection step in order to de-select the attributes that do not contribute to class separation and/or to select other attributes that were not previously used, (2) try to improve the classification result by manipulation of the parameters 138 (the type of parameter and its effect depends on the classification method being used), and/or (3) declare the classification "finished" and save the intermediate results or part of the intermediate results to storage.

When step 6 ("evaluate distribution separation" 170 of the resultant attribute data set) is complete, the operator at workstation 100 of figures 14 and 15 will have another opportunity to indicate whether another "manual inspection" of the input attributes 134, 136 and the parameters 138, relative to its corresponding cross plot in attribute space, is necessary via the manual inspection block 156 in figure 26.

Assume now that another such manual inspection of the input attributes 134, 136, relative to the resultant cross plot in attribute space, is not necessary, and, as a result, a "no" output 156a is generated from the manual inspection block 156 in figure 26.

In figure 26, if a "no" output 156a is generated from the manual inspection block 156, the input attributes 134, 136 and the parameters 138 of figure 24 were "tested" by clustering "some" of the attributes 134, 136 (called "inspection data") in attribute space (similar to the attribute

5   space 114 shown in figure 18), and determining if that "inspection data" clustered acceptably in attribute space (114 of figure 18). In addition, the input attributes 134, 136 were further tested by selecting some of the points within a cluster on the "cross plot in attribute space" (called "further inspection data"), generating a corresponding "attribute data set"

10   (54a, 54b in figure 13) from that further inspection data, and evaluating the distribution of the attribute data on the resultant "attribute data set" relative to the further inspection data in the "cross plot in attribute space".

15   Alternatively, in figure 26, the operator could have tested the clustering of "some" of the attributes 134, 136 (the inspection data 158) in attribute space, determined if the attribute "inspection data" clustered in an acceptable fashion in attribute space (114), and then exited, at this point, via the "manual inspection" block 164 in figure 26 by answering

20   "no" to the question "manual inspection?" 164. In that case, a "yes" answer to the question "satisfactory separation" 170 in figure 26 would be generated, indicating that the separation of the clusters in the "cross plot in attribute space" (114) were acceptable and satisfactory.

25   In figure 26, if a "no" output 156a is generated from the manual inspection block 156, then, the clusters in the "cross plot in attribute space" relative to the inspection data on an "attribute data set" are acceptable, the distribution on a resultant "attribute data set" relative to inspection data on the "cross plot in attribute space" are also acceptable,

30   and, as a result, the classification algorithm 172 in figure 26 can now be applied (i.e., the block "apply classification algorithm" 172 block of code of figure 26 can now be executed).

In figure 29, when the "apply classification algorithm" 172 block of code of figure 26 is executed, while using the "parameters" 138 of figure 25, all of the plurality of input attributes (a_x, b_x) 177 of figure 29 (134, 136 of figure 25) on the "attribute data sets" of figure 29 will now be clustered in

5   the "cross plot in attribute space" 179 of figure 29 (not just "some" of the input attributes which were previously called the "inspection data"). A different label, such as color, will now be assigned to each cluster in attribute space. In addition, after the different labels, such as colors, are assigned to the respective clusters in attribute space 179, a "final

10   resultant class data set" 130 (i.e., the "Classified Result" 130 in figure 23) is generated, wherein each of the respective plurality of points on the "a"/"b" composite "attribute data set" (of figure 12) is colored with its respective color, that color being obtained from one of the clusters on the "cross plot in attribute space" 179 of figure 29.

15

The above discussion with reference to figures 23 through 29 discussed the "Un-supervised" seisclass software 100b1 of figure 22. The following discussion with reference to figures 30 through 43 will now discuss the "Supervised" seisclass software 100b2 of figure 22.

20

Recall that the "Un-supervised" seisclass software 100b1 of figure 22 is used when the "classes" of the input attribute data on the attribute data sets 54a, 54b of figure 13, disposed within the attribute data sets 46 of figure 14, are not known. However, the "Supervised" seisclass software

25   100b2 of figure 22 is used when such "classes" are known (where the word "classes" means the class or category to which the data in the attribute data sets 46 belong).

Referring to figures 30 through 43, the "Supervised" seisclass software

30   100b2 of figure 22 is discussed in the following paragraphs with reference to figures 30 through 43.

34

In figure 30, the structure of the Supervised seisclass software 100b2 includes the following. A supervised classification block of code 186 will receive six (6) different blocks of input data: (1) the input attribute data 134, (2) the well supplement attribute data 136, (3) a group of attribute

5   data known as "training/validation known class at position" 182, (4) another group of attribute data known as "well training/validation known class at borehole" 184, (5) the "parameters controlling the classification process" 138, and (6) the "classes in use - manually defined class types" 188. In response to the above six (6) blocks of data, the

10   supervised classification code 186 will be executed, and a "cross plot in attribute space" will be generated. Different labels, such as colors, are assigned to each cluster in the "cross plot in attribute space". Then, the "classified result" 130 block of code will generate a class data set 130, similar to the class data set 130 shown in figure 21. The "auxiliary

15   results QC measures" code 142 will be executed in an identical manner as described above with reference to figure 24.

In figure 31, the above referenced input data entitled "classes in use - manually defined class types" 188 of figure 30 will be defined and

20   discussed below with reference to figure 31.

The input attribute data on the attribute data sets of figure 13 and in the "attribute data sets" 46 of figure 14 include a plurality of "subsets" of data, and different ones of those "subsets" of data may belong to different

25   "classes". Those different "classes" are illustrated in figure 31. The different "classes" of figure 31 are categories to which the attribute data in the attribute data sets 46 belong. Some of the attribute data in the attribute data sets 46 belong to one class or category, and other attribute data in the attribute data sets 46 belong to another class or category. In

30   figure 31, there are four (4) such classes or categories illustrated: (1) the "litho-class family" 174, (2) the "pore-fill class family" 176, (3) the "stratographic class family" 178, and (4) the "unsupervised class family" 180. The litho-class family 174 includes: sand 174a, shale 174b,

limestone 174c, sandy shale 174d, and shaly sand 174e, and the sand
174a further includes sand type "A" 174a1 and sand type "B" 174a2.
The pore-fill class family 176 includes: gas 176a and fluid 176b, and the
fluid 176b further includes oil 176b1 and brine 176b2. The

5   stratographic class family 178 includes the following: parallel 178a,
hummocky 178b, type A 178c, and type B 178d. The unsupervised class
family 180 includes: C1 180a, C2 180b, and C3 180c (when attribute
data in the attribute data sets are not known to belong to any particular
class, the unsupervised class family 180 is selected by the operator).

10

The operator at workstation 100 of figures 14 and 15 will view the
"attribute data sets" 46 and he will attempt to categorize the individual
input attribute data set forth within the "attribute data sets" 46 of figure
14 by grouping sub-sets of that data into the one or more of the classes

15   illustrated in figure 31. That is, the operator defines a set of "classes" to
be used in the classification, simultaneously requiring that the classes or
its children be represented in the training and validation step to be
discussed below. The selection, by the operator, of the "classes", into
which the operator classifies the data, is an interpretative decision. In

20   each specific setting, there should be a minimum justification for
selecting a class, e.g., through evidence that the transition from "class A"
to "class B" correlates with an observable change in attribute values that
again have a physical explanation.

25   In figure 30, the structure of the "Supervised" seisclass software 100b2
of figure 22 will again be discussed with reference to figure 30.

Recall that the Seisclass software 100b of figure 14 and 22 includes two
parts: the "Un-supervised" seisclass software 100b1, and the

30   "Supervised" seisclass software 100b2. Compare figure 23
("Unsupervised" seisclass software 100b1) with figure 30 ("supervised"
seisclass software 100b2). In figure 30, a "supervised classification"
block 186 receives certain "inputs", and, responsive thereto, a "classified

36

result (class dat⬤t)" 130 is generated. The quality⬤the classified

result 130 (i.e., the quality of the class data set 130) is checked via the

"auxiliary results QC measures" block 142 in figure 30, as discussed

above with reference to figures 23 and 24. In figure 30, the "input

5   attributes" 134 and the "well supplement attributes" 136 and the

"parameters controlling the classification process" 138, discussed above

with reference to figure 23, are also provided as "inputs" to the

"supervised classification" block 186. However, in addition to the "input

attributes" 134 and the "well supplement attributes" 136 and the

10   "parameters controlling the classification process" 138, there are three

additional "inputs" being provided to the "supervised classification" 186

block of figure 30: (1) "training/validation - known class at position" 182,

(2) "well training/validation - known class at borehole" 184, and (3)

"classes in use - manually defined class types" 188.

15

In figure 30, recall from figure 10 that the "input attributes" 134

represent, for example, the following amplitude and frequency attributes:

(a6, f2) 74, 76, (a3, f5) 78, 80, (a5, f3) 82, 84, and (a4, f6) 86, 88. In

addition, the "well supplement attributes" 136 represent, for example,

20   the following amplitude and frequency attribute: (a2, f4) 90, 92. Recall

also, in connection with the "parameters controlling the classification

process" 138, that the seisclass software 100b offers a selection of

classification methods, and each such method has its own particular

"parameters" which are closely related to the particular method. In

25   addition, in figure 30, the "classes in Use - manually defined class types"

188 have already been discussed above with reference to figure 31 and

include the following classes: the litho-class family 174, the pore fill class

family 176, the stratographic class family 178, and the unsupervised

class family 180.

30

However, in figure 30, in addition to the input data entitled "input

attributes" 134 and "well supplement attributes" 136 and "parameters"

138 and "classes" 188, there are two other blocks of input data being

provided as input to the supervised classification code 186 of figure 30: (1) "training/validation - known class at position" block 182, and (2) "well training/validation - known class at borehole" block 184.

5   In figures 32 through 37, the input data in figure 30 entitled "training/validation - known class at position" 182 and "well training/validation - known class at borehole" 184 of figure 30 will be discussed below with reference to figures 32 through 37.

10   In the following paragraphs, reference is made to the "a" attribute and the "b" attribute, or the attribute data set "a" or the attribute data set "b". The "a" attribute can, for example, be one of the "amplitude" attributes of the attribute data set 54a of figure 13, and the "b" attribute can, for example, be one of the "frequency" attributes of the attribute data set
15   54b of figure 13.

     In figures 32 and 33, an attribute data set "a" 190 and attribute data set "b" 192 have two boreholes 194 and 196 disposed through the attribute data sets. These boreholes 194, 196 have been logged using the logging
20   techniques discussed above with reference to figure 2 and therefore the "classes" 188 for both said boreholes 194, 196 have been determined. Let us assume, for purposes of this discussion, that the "class" 188 for borehole 194 is the same as the "class" 188 for borehole 196. The reason for this assumption will become apparent from the following discussion.
25   The attribute data set "a" 190 contains "a" attributes, such as frequency "f", and the attribute data set "b" 192 contains "b" attributes, such as amplitude "a".

     In figure 32, the "a" attributes located inside borehole 194 and located
30   around the borehole 194 are illustrated. By way of definition, the "A1" attribute located inside the borehole 194 is labelled "borehole attribute" and the "a" attributes located inside and around the borehole 194 (located within an outer boundary 194a) are labelled "position

38

attributes". In figure 32, therefore, the "a" "position" attributes located within the boundary 194a are as follows: a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12. On the other hand, in figure 32, the "a" "borehole" attribute located inside the borehole 194 itself is "A1". The "a" "position" attributes "a1" through "a12" including "a7" were obtained from the seismic operation illustrated in figure 1, whereas the "a" "borehole" attribute "A1" was obtained from the logging operation illustrated in figure 2.

In figure 33, the "b" attributes located inside borehole 194 and located around the borehole 194 are illustrated. By way of definition, the "B1" attribute located inside the borehole 194 is labelled a "borehole attribute" and the "b" attributes located inside and around the borehole 194 (located within an outer boundary 194b) are labelled "position attributes". In figure 33, therefore, the "b" "position" attributes located within the boundary 194b are as follows: b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12. On the other hand, in figure 33, the "b" "borehole" attribute located inside the borehole 194 itself is "B1". The "b" "position" attributes "b1" through "b12" including "b7" were obtained from the seismic operation illustrated in figure 1, whereas the "b" "borehole" attribute "B1" was obtained from the logging operation illustrated in figure 2.

Let us now declare the "a" and "b" attributes shown in figures 32 and 33 to be "training data", short for "training attribute data".

In figure 34, a table is illustrated which specifically identifies the "borehole training data" to be "(A1, B1)", where A1 and B1 are the "borehole" attributes, and specifically identifies the "position training data" to be (a1, b1), (a2, b2), (a3, b3), (a4, b4), (a5, b5), (a6, b6), (a7, b7), (a8, b8), (a9, b9), (a10, b10), (a11, b11), and (a12, b12), where (a1, b1) through (a12, b12) are "position" attributes.

39

In figure 35, the attributes located inside borehole 196 and located around the borehole 196 are illustrated. By way of definition, the "A2" attribute located inside the borehole 196 is labelled "borehole attribute" and the "a" attributes located inside and around the borehole 196 (i.e.,

5   located within an outer boundary 196a) are labelled "position attributes". In figure 35, therefore, the "a" "position" attributes located within the boundary 196a are as follows: a13, a14, a15, a16, a17, a18, a19, a20, a21, a22, a23, a24. On the other hand, in figure 35, the "a" "borehole" attribute located inside the borehole 196 itself is "A2". The "a" "position"

10   attributes "a13" through "a24" including "a19" were obtained from the seismic operation illustrated in figure 1, whereas the "a" "borehole" attribute "A1" was obtained from the logging operation illustrated in figure 2.

15   In figure 36, the "b" attributes located inside borehole 196 and located around the borehole 196 are illustrated. By way of definition, the "B2" attribute located inside the borehole 196 is labelled a "borehole attribute" and the "b" attributes located inside and around the borehole 196 (i.e., located within an outer boundary 196b) are labelled "position attributes".

20   In figure 36, therefore, the "b" "position" attributes located within the boundary 196b are as follows: b13, b14, b15, b16, b17, b18, b19, b20, b21, b22, b23, b24. On the other hand, in figure 36, the "b" "borehole" attribute located inside the borehole 196 itself is "B2". The "b" "position" attributes "b13" through "b24" including "b19" were obtained from the

25   seismic operation illustrated in figure 1, whereas the "b" "borehole" attribute "B2" was obtained from the logging operation illustrated in figure 2.

Let us now declare the "a" and "b" attributes shown in figures 35 and 36

30   to be "validation data", short for "validation attribute data".

40

In figure 37, a table is illustrated which specifically identifies the "borehole validation data" to be "(A2, B2)" and specifically identifies the "position validation data" to be (a13, b13), (a14, b14), (a15, b15), (a16, b16), (a17, b17), (a18, b18), (a19, b19), (a20, b20), (a21, b21), (a22, b22), (a23, b23), and (a24, b24).

In figures 34 and 37, referring alternately to the tables of figures 34 and 37, the "training/validation - known class at position" 182 of figure 30 would include the following attribute data: (a1, b1), (a2, b2),..., (a12, b12) for training, and (a13, b13), (a14, b14),...,(a24, b24) for validation. Similarly, the "well training/validation - known class at borehole" 184 of figure 30 would include the following attribute data: (A1, B1) for training and (A2, B2) for validation.

Referring to figure 38, the structure of the "supervised classification" block 186 of figure 30 is illustrated.

In figure 38, the input data blocks, which are input to the "supervised classification" block 186 include the following: (1) "input attribute selection" 134, 136 consisting of the "input attributes" 134 and the "well supplement attributes" 136, (2) the "training/validation" 182, 184 consisting of the "training/validation - known class at position" 182 and the "well training/validation - known class at borehole" 184, (3) the "parameters controlling the classification process" 138, and (4) the "classes in use - manually defined class types" 188. Each of these input data blocks have been discussed in detail in the foregoing paragraphs of this specification.

In figure 38, in connection with the "classes" block 188, it is important to understand that the operator will attempt to determine into which "class" (of figure 31) each of the "input attributes" 134, 136 belong, and into which "class" (of figure 31) each of the "training/validation" attributes 182, 184 belong. When those "classes" for each of the "input attributes"

41

134, 136 and each of the "training/validation" attributes 182, 184 are determined, those "classes" will collectively comprise and constitute the "classes" block 188 in figure 38.

5   In figure 38, the input attributes selection 134, 136; and the training/validation 182, 184; and the parameters 138; and the classes 188 are each: (1) saved and stored, via the "save input" block 206 in figure 38, and (2) input to the "training of the classifications" block 208 via the element 210 in figure 38.

10

In figure 38, the "training of the classifications" block 208 is executed; the parameters 138 used during the execution of the "training of the classifications" 208 block of code are saved via the "save parameters" block 210; and the question "is validation successful?" is asked, via

15   block 212.   If the above referenced validation is not successful, a "no" output 213 is generated from the "validation successful" block 212 in figure 38, and the "edit block" 214 in figure 38 is executed.   When the edit block 214 is executed, the operator can now change the "input data" and try again.   In the edit block 214 of figure 38, the operator can either:

20   (1) edit the "training/validation" 182, 184 via the "edit training/validation" block 214a, and/or (2) edit the "classes" 188 via the "select/redefine classes" block 214b, and/or (3) edit the "input attributes selection" 134, 136 via the "edit input attribute selections" block 214c, and/or (4) edit the "parameters" 138 via the "edit parameters" block

25   214d.   In figure 38, if the operator will "edit training/validation" of block 214a, he will change and/or modify the "training/validation" attributes 182, 184 being input to the "training of the classifications" block 208. Alternatively, if the operator will "select/redefine classes" block 214b, he will change and/or modify the "classes" 188 which were originally

30   selected by the operator.   If the operator will "edit input attribute selection" block 214c, he will change and/or modify the "input attributes selection" 134, 136.   If the operator will "edit parameters" block 214d, he will change and/or modify the "parameters" 138 in figure 38 which are

42

input to the "training of the classifications" block 208. When the
operator edits the training/validation or the classes or the input
attributes or the parameters via the edit block 214 in figure 38, the
"training of the classifications" block 208 is again re-executed as

5   previously discussed, the parameters are re-saved via the "save
parameters" block 210, and the following question is asked once again:
"is validation successful" 212.   If the validation is successful, the "apply
the trained classification and determine quality indication", block 216 in
figure 38, is executed.   When the "apply the trained classification and

10 determine quality indication" block 216 is executed, all of the input
attribute data (including the training and validation attribute data) will
be clustered in attribute space, in the same manner as illustrated in
figure 18 of the drawings. The clustering in attribute space is studied to
determine if "that clustering" is acceptable, via the "quality acceptable"

15 block 218 in figure 38. The results achieved from "that clustering" are
temporarily saved in the "save results" block 220. If the quality of "that
clustering" is acceptable, the results from "that clustering" are
permanently saved in the "save results" block 220, and the "classified
result" 130 (i.e., the class data set 130 of figure 21) is generated, block

20 222 of figure 38 and block 130 of figure 30.   If the quality of "that
clustering" is not acceptable, a "no" output from the "quality acceptable"
block 218 will be input to the "edit block" 214, wherein the
training/validation attributes can be edited, 214a, and/or the classes
can be redefined, 214b, and/or the input attribute selection can be

25 edited, 214c, and/or the parameters can be edited, 214d, at which point,
the "training of the classifications" block 208 and the "validations
successful" blocks 212 can be re-executed in hopes of achieving an
acceptable quality output from the "quality acceptable" block 218. The
flowcharts of figures 38 and 39 will be better understood following a

30 reading of the remaining parts of this specification.

Referring to figure 39, the structure of the "training of the classifications"
block of code 208 of figure 38 is illustrated.

43

In figure 39, starting with the element 210, the "input data" (which includes the input attributes 134, 136, the training/validation attribute data 182, 184, the parameters 138, and the classes 188) are input to the

5   element 210. An operator sitting at workstation 100 of figure 15 has an opportunity, at this point, to inspect the input attributes 134, 136. If a manual inspection of the input attributes 134, 136 is necessary, the operator at the workstation 100 will indicate that such an inspection is necessary thereby triggering the execution of the first "manual

10   inspection" block 224 in figure 39 (note that there are two "manual inspection" blocks in figure 39, a first "manual inspection" block 224, and a second "manual inspection" block 232). Assume that the operator indicated that a manual inspection of the input attributes 134, 136 is necessary and that, as a result, the "yes" output from the first "manual

15   inspection" block 224 in figure 39 is generated. In figure 39, if the "yes" output from the first "manual inspection" block 224 of figure 39 is generated, the following portions of the "training of the classifications" code 208 in figure 39 will be executed:

20   (Step 1) "select inspection data" 226,
      (Step 2) "generate plot in attribute space" 228, and
      (Step 3) "evaluate distribution/separation" 230.

      In figure 40, during step 1, inspection data 189, consisting of a subset of

25   the input attributes (ax, bx) on the attribute data sets 54a and 54, are selected from among all the other input attributes (ax, bx) on the attribute data sets 54a, 54b in figure 40 (block 226 of figure 39). During step 2, using that inspection data 189, a "cross plot in attribute space" 191 is generated in the same manner as discussed above with reference

30   to figure 18 (block 228 of figure 39). During step 3, the distribution/separation of the two clusters 191a, 191b on the cross plot in attribute space 191 is evaluated (block 230 of figure 39).

In figure 39, if the "yes" output from the second "manual inspection" block 232 in figure 39 is generated, the following additional portions of the "training of the classifications" code 208 in figure 39 will be executed:

5   (Step 4) "select inspection data" 234,

(Step 5) "generate data set plot" 236, and

(Step 6) "evaluate distribution" 238

In figure 41, during step 4, inspection data 193 is selected, the

10   inspection data 193 consisting of a subset of the points on the "cross plot in attribute space". In our example in figure 41, a cluster 193 on the "cross plot in attribute space" has been selected to represent the inspection data 193 (block 234 of figure 39). During step 5, the inspection data 193 on the "cross plot in attribute space" of figure 41 is

15   then used to generate a data set plot 195 on the attribute data sets 54a, 54b (block 236 of figure 39). During step 6, the distribution of the set of points on the data set plot 195 on the attribute data sets 54a, 54b is then evaluated (block 238 of figure 39).

20   In figure 39, when step 6 ("evaluate distribution" block 238 of figure 39) is complete, the operator at workstation 100 of figures 14 and 15 will have another opportunity to indicate whether another "manual inspection" of the input attributes 134, 136 is necessary via the manual inspection block 224 in figure 39. Assume now that another such

25   manual inspection of the input attributes 134, 136 is not necessary, and, as a result, a "no" output 224a is generated from the manual inspection block 224 in figure 39. In response to the "no" output 224a from the manual inspection block 224, the "training of the classification" code 242 in figure 39 is executed.

30

In figures 39 and 30, when the "training of the classification" code 242 of figure 39 is executed, the attribute data stored under the label

"training/validation - known class at position" 182 of figure 30 is first used to generate a "cross plot in attribute space".

In figures 34 and 37, referring initially to figure 34, the "training-position attribute data" of figure 34 [(a1, b1), (a2, b2),..., and (a12, b12)] is first plotted in "attribute space" thereby generating a first set of points on the "cross plot in attribute space". We know, from our previous assumption, that the figure 31 class of the "training-position attribute data" of figure 34 is the same as the figure 31 class of the "validation-position attribute data" of figure 37. Therefore, the "validation-position attribute data" of figure 37 should cluster near or adjacent to the "training-position attribute data" of figure 34 on the "cross plot in attribute space". As a result, in figure 37, the "validation-position attribute data" of figure 37 [(a13, b13), (a14, b14),..., and (a24, b24)] is also plotted in "attribute space" alongside the "training-position attribute data" of figure 34 [(a1, b1), (a2, b2),..., and (a12, b12)] thereby generating/plotting a second set of points on the same "cross plot in attribute space".

In figures 39 and 30, when the "training of the classification" code 242 of figure 39 is further executed, the attribute data stored under the label "well training/validation - known class at borehole" 184 of figure 30 is then used to plot another set of points on the same "cross plot in attribute space".

In figures 34 and 37, referring initially to figure 34, the "training-borehole attribute data" of figure 34 "(A1, B1)" is first plotted in the same "attribute space" thereby generating a third point on the same "cross plot in attribute space". We know, from our previous assumption, that the figure 31 class of the "training-borehole attribute data" of figure 34 is the same as the figure 31 class of the "validation-borehole attribute data" of figure 37. Therefore, the "validation-borehole attribute data" of figure 37 should cluster near or adjacent to the "training-borehole attribute data" of figure 34 on the same "cross plot in attribute space". As a result, in

figure 37, the "validation-borehole attribute data" of figure 37 "(A2, B2)" is also plotted in "attribute space" alongside the "training-borehole attribute data" of figure 34 "(A1, B1)" thereby generating/plotting a second point on the same "cross plot in attribute space".

In figure 42, regarding the execution of the "training of the classification" code 242 of figure 39, when the "training-position attribute data" of figure 34 and the "validation-position attribute data" of figure 37 and the "training-borehole attribute data" of figure 34 and the "validation-borehole attribute data" of figure 37 are plotted in the same "cross plot in attribute space", as discussed above, that same "cross plot in attribute space" is illustrated in figure 42. Note, in figure 42, that the "validation data cluster" 199 does, in fact, cluster adjacent to the "training data cluster" 197. The "validation data cluster" 199 should cluster adjacent to the "training data cluster" 197 because the figure 31 class of the training data of figure 34 was assumed to be the same as the figure 31 class of the validation data of figure 37.

In figures 38 and 42, regarding the "validation successful" block 212 of figure 38, since the "validation data cluster" 199 of figure 42 did, in fact, cluster adjacent to the "training data cluster" 197 in figure 42, the validation was successful, and, as a result, a "yes" output is generated from the "validation successful" block 212 of figure 38. In figure 38, the next block of code to be executed is the "apply the trained classification determine quality indication" block 216 in figure 38. In order to under this block of code (216 in figure 38), refer now to figure 43.

In figure 43, during the execution of the "apply the trained classification..." block 216 of figure 38, all of the attribute data 201 on the attribute data sets 54a, 54b in figure 43 will be plotted on a "cross plot in attribute space", using the techniques discussed above with reference to figures 17 through 21. When that "cross plot in attribute space" is generated, the "cross plot in attribute space" 203 illustrated in

figure 43 will be generated. The four clusters 203a through 203d in the cross plot 203 of figure 43 will each be assigned a different label, such as color. The attribute data inside each cluster will each be assigned a label, such as color.

5

In figure 38, assuming that the operator at workstation 100 of figure 15 believes that the quality of the "cross plot in attribute space" 203 of figure 43 is acceptable, a "yes" output is generated from the "quality acceptable" block 218 of figure 38. The next block of code to be executed

10   is the "classified result" block of code 130 of figure 30 (or block 222 of figure 38).

In figures 30 and 44, when the "classified result" block 130 of figure 30 is executed, the "cross plot in attribute space" 203 of figure 44 (and figure

15   43) is used to generate a "class data set" 130 in figure 44, using the same techniques discussed above with reference to figures 18 through 21.

In figure 44A, now that we know how to generate the "class data set" 130 in figure 44, by using those same techniques, a 3D cube can be

20   generated, such as the 3D cube illustrated in figure 44A. In figure 44A, three different sub-surfaces have three different class data sets associated therewith. That is, a first class data set 130a reflects a first set of characteristics of a first sub-surface 14a, a second class data set 130b reflects a second set of characteristics of a second sub-surface 14b,

25   and a third class data set 130c reflects a third set of characteristics of a third sub-surface 14c in an earth formation. The class data set 130 of figure 44 is designed for 2D horizon/map data; however, the Seisclass software 100b of the present invention is also suited for classification of 3D cubes, such as the 3D cube illustrated in figure 44A, 4D time-lapse

30   cubes (where time is the fourth dimension), or 5D pre-stack time-lapse cubes.

A functional operation of the "Seisclass Software" 100b of the present
invention, when executed by the workstation processor 100a of figure 14,
is set forth in the following paragraphs with reference to figures
1 through 45 of the drawings.

5

A seismic operation is performed on an earth formation in the manner
illustrated in figure 1 thereby producing a seismic data output record 24.
In addition, a well logging operation is performed in the same earth
formation in the manner illustrated in figure 2 thereby producing a well

10   log output record 32. The seismic data output record 24 and the well log
output record 32 combine to represent the data received 42 of figure 5;
and, in figure 6, the data received 42 is used, by the computer 44 of
figure 6, to generate the attribute data sets 46. For example, following
the seismic operation of figures 7, 8, and 9, the computer 44 of figure 6

15   will generate the attribute data set 46 of figure 13. The attribute data
sets 46 of figure 13 includes a first "attribute data set" 54a and a second
"attribute data set" 54b. The first attribute data set 54a in figure 13 will
include a multitude of "a" attributes, such as "amplitude (a)" values. The
second attribute data set 54b in figure 13 will include a multitude of "b"

20   attributes, such as "frequency (f)" values. Actually, as shown in figure
10, the "attribute data sets" 46 of figure 13 actually include the "input
attributes" 134 and the "well supplement attributes" 136.

In figures 14 and 15, the attribute data sets 46 of figures 10 and 13,

25   which include the input attributes 134 and the well supplement
attributes 136, are provided as "input data" to the workstation 100 of
figures 14 and 15. However, the "seisclass software 100b" in figure 15,
initially stored on a storage medium 102, such as a CD Rom 102, is
loaded into the workstation 100 of figure 15 and is stored into the

30   workstation memory 100b. When the seisclass software 100b is stored
in the workstation memory of figure 14, and when the workstation
processor 100a of figure 14 executes the seisclass software 100b, the
workstation processor 100a will perform the following overall function, as

49

noted in figure 16. (1) produce a cross plot in attribute space from the attribute data sets, block 112a, (2) subdivide the cross plot into a plurality of zones comprised of points which correspond to other points on the attribute data sets where each zone on the cross plot has a

5 different class association than any other zone which is indicated by a different label, such as a different color, block 112b, and (3) produce a class data set 130 comprised of a plurality of points on the attribute data sets where each point has a label, such as color, depending on its cluster/class association on the cross plot in attribute space, block 112c.

10 The "cross plot in attribute space" is illustrated, in a simple example, by the cross plot 114 of figure 18, and the class data set is illustrated, in the simple example, by the class data sets 130 of figures 20 and 21.

When the seisclass software 100b of figure 14 is executed, the operator

15 at the workstation 100 must first examine the "attribute data", included within the attribute data sets 46 (including the input attributes 134 and the well supplement attributes 136), to determine whether that "attribute data" in the attribute data sets 46 can be conveniently classified into one or more of the "classes" which are illustrated in

20 figure 31.

If that "attribute data" cannot be classified into one or more of the classes of figure 31, the "unsupervised" seisclass software 100b1 of figure 22 is executed. The "unsupervised" seisclass software 100b1 is

25 discussed below with reference to figures 23 through 29 of the drawings.

However, if that "attribute data" can, in fact, be classified into one or more of the classes of figure 31, the "supervised" seisclass software 100b2 of figure 22 is executed. The "supervised" seisclass software

30 100b2 is discussed below with reference to figures 30 through 44 of the drawings.

50

In figure 23, if the "attribute data" in the attribute data sets 46 cannot be classified into one or more of the classes of figure 31 and the "unsupervised" seisclass software 100b1 of figure 22 is executed by the workstation processor 100a of figure 14, the unsupervised classification code 132 of figure 23 will begin execution in response to the input attributes 134 and the well supplement attributes 136 of the attribute data sets 46 and the parameters controlling the classification process 138. When the unsupervised classification code 132 has completed its execution, the classified result 130 (i.e., a class data set similar to the class data set 130 of figure 21) is generated. The auxiliary results/QC measures code 142 of figure 23 will assess the quality of the class data set/classified result 130.

In figure 25, recalling that the input attributes 134 and the well supplement attributes 136 comprise the input attributes 134, 136 of figure 25, when the unsupervised classification code 132 of figures 23 and 25 is executed by the workstation processor 100a of figure 14, the "input data", consisting of the input attributes 134, 136 and the parameters 138, are initially saved via the save input block 144 in figure 25 and the "input data" are received by the "apply classification and determine quality indicators" code 146 of figure 25. This "apply classification..." code 146 will then "apply the classification" and then "determine quality". If the "quality" is not acceptable, via block 152 of figure 25, the input attributes 134, 136 can be edited, and the parameters 138 can be edited, via the edit block 154 of figure 25, and then the "apply classification and determine quality indicators" code 146 will be re-executed. If the "quality" is acceptable, via block 152 of figure 25, the classified result/class data set 130 is then generated. A sample class data set 130 is shown in figures 20 and 21.

As noted above, the "apply classification and determine quality indicators" code 146 of figure 25 will "apply the classification" and then "determine quality". Figure 26 illustrates how the "apply classification

51

and determine quality indicators" code 146 "applies the classification" and "determines quality".

In figures 26, the input attributes 134, 136 and the parameters 138 are received by the "apply classification..." code 146; and, responsive thereto, the "apply classification..." code 146 will: (1) perform its first function to "determine quality" by executing code blocks 156 through 170 in figure 26, and then, when the quality is acceptable, the "apply classification..." code 146 (2) will perform its second function to "apply the classification" by executing code block 172 in figure 26.

In figures 26 and 27, when the "apply classificaton..." code 146 of figure 26 "determines quality" by executing the code blocks 156 through 170 in figure 26, the operator selects the inspection data 173 of figure 27 which represents a subset of the attribute data sets 54a, 54b of figure 13 (block 158 of figure 26), generates a "cross plot in the attribute space" consisting of a pair of clusters of points 181a and 181b in the attribute space of figure 27 (block 160 of figure 26), and evaluates the distribution and separation of the clusters 181a and 181b on the "cross plot in attribute space" in figure 27 (block 162 of figure 26).

In figures 26 and 28, if a further manual inspection of the input attributes 134, 136 is necessary (block 164 of figure 26), the operator will select the cluster 185 in the attribute space of figure 28 as "inspection data" (block 166 of figure 26), generate the attribute data set plot 187 of figure 28 (block 168 of figure 26), and evaluate the distribution inherent in the attribute data set plot 187 of figure 28 (block 170 of figure 26). If the distribution or separation of the attribute data on the attribute data set plot 187 of figure 28 is satisfactory, no further manual inspection of the attribute data 134, 136 is necessary (see line 156a exiting from the "no" output from the "manual inspection" block 156 of figure 26). At this point, the "apply classification algorithm" 172 of figure 26 will begin execution.

In figure 29, when the "apply classification algorithm" 172 of figure 26
begins execution, all of the input attributes 134 and the well supplement
attributes 136 of the attribute data sets 54a, 54b of the attribute data
sets 46 of figure 29 (including the inspection data 173 of figure 27) are
used to generate a "cross plot in attribute space" 179 and a "final
crossplot in attribute space" 179 is produced which is similar to the
cross plot 114 shown in figure 18. The "cross plot in attribute space"
179 is generated from the attribute data sets 54a, 54b of figure 29 using
the same techniques discussed above with reference to figures 12, 13,
17, and 18. In figure 25, the quality of that "final cross plot" 179 is
assessed, via the "quality acceptable" block 152 of figure 25. If the
quality of that "final cross plot" is acceptable, the "final cross plot" 179
data/results are saved via the "save results" block 150 in figure 25, and
the "final cross plot" 179 is used to plot the final classified result
130/final class data set 130 in figure 23 and 25 similar to the class data
set 130 shown in figures 20 and 21, using the same techniques
discussed above with reference to figures 18, 20, and 21. The "auxiliary
results QC measures" code 142 of figure 23 will assess the quality of that
final class data set 130.

In figure 30, however, if the "attribute data" in the attribute data sets 46
can, in fact, be classified into one or more of the classes of figure 31, the
"supervised" seisclass software 100b2 of figure 22 is executed by the
workstation processor 100a of figure 14. When the "supervised"
seisclass software 100b2 of figure 30 is executed by the workstation
processor 100b2, the supervised classification code 186 of figure 30 will
receive certain "input data". When the supervised classification code 186
of the supervised seisclass software 100b2 is executed, the supervised
classification code 186 will respond to that "input data" by generating the
classified result 130/class data set 130. A sample class data set 130 is
illustrated in figure 21. The "auxiliary results QC measures" code 142 of

53

figure 30 will assess the quality of the classified result 130/class data set 130 generated by the supervised classification code 186.

The "input data", received by the supervised classification code 186 of figure 30, includes the "attribute data sets" 46, the "parameters" controlling the classification process 138, and the "classes in use" 188. The "attribute data sets" 46 of figure 30 include four (4) parts: (1) the input attributes 134 of figure 10, (2) the well supplement attributes 136 of figure 10, (3) the training/validation known class at position attributes 182, and (4) the well training/validation known class at borehole attributes 184. The input attributes 134 and the well supplement attributes 136 were discussed above with reference to figure 10. The "training/validation known class at position" attributes 182 are represented by the attributes shown under the "position" column in the tables of figures 34 and 37. The "well training/validation known class at borehole" attributes 184 of figure 30 are represented by the attributes shown under the "borehole" column in the tables of figures 34 and 37. The "classes in use" 188 in figure 30 are defined by the operator; that is, in order to define the "classes in use" 188, the operator must classify each subset of the attribute data in the attribute data sets 46 of figure 30 to determine into which class, of the classes of figure 31, each subset of the attribute data belong. When this task is complete, the "classes in use - manually defined class types" block 188 of figure 30 are defined for the attribute data present in the attribute data sets 46. For example, in figure 30, since the "attribute data sets" 46 include the "input attributes" 134 and the "well supplement" attributes 136 and the "training/validation known class at position" attributes 182 and the "well training/validation known class at borehole" attributes 184, the operator must analyze all the attribute data in the attribute data sets (134, 136, 182, 184) of figure 30 and, for each "subset of attribute data" of the attribute data sets 46, the operator must determine into which class (of the classes of figure 31) each such "subset" belongs. For example, one "subset" may belong to the "sand" class 174a of figure 31 and another

54

"subset" may belong to the "brine" class 176b2 of figure 31. When this task is complete, the "classes in use - manually defined class types" 188 block of "input data" of figure 30 is fully defined.

5   The supervised classification code 186 of figure 30 is illustrated in greater detail in figure 38. In figure 38, when the supervised classification code 186 is executed, the training of the classifications code 208 will receive the above referenced "input data". In figure 39, the training of the classifications code 208 will: select the inspection data
10   189 of figure 40 (block 226 of figure 39), generate a cross plot 191 in the attribute space of figure 40 (block 228 of figure 39), and evaluate the distribution of the clusters 191a and 191b in the attribute space 191 of figure 40 (block 230 of figure 39). Then, if a further manual inspection of the attribute data in the "input attributes selection" 134, 136 is
15   necessary (block 232 of figure 39), the training of the classifications code 208 will: select the inspection data 193 in the "cross plot in attribute space" of figure 41 (block 234 of figures 39 and 41), generate an attribute data set plot 195 of figure 41 (block 236 of figures 39 and 41), and evaluate the distribution in the attribute data set plot 195 (block 238 of
20   figures 39 and 41). If the distribution of the attribute data on the attribute data set plot 195 of figure 41 appears to be satisfactory (block 238 of figure 39), the training of the classifications code 208 of figure 39 may then execute the "training of the classifications" code 242 in figure 39. The execution of the "training of the classifications" code 242 of
25   figure 39 will involve: (1) the clustering of the "training-position" data of figure 34 in attribute space, clustering the "validation-position" data of figure 37 in the same attribute space, and determining if the "validation-position" data of figure 37 also clusters adjacent to the "training-position data of figure 34 in the same attribute space, and (2) the clustering of the
30   "training-borehole" data of figure 34 in attribute space, clustering the "validation-borehole" data of figure 37 in the same attribute space, and determining if the "validation-borehole" data of figure 37 also clusters adjacent to the "training-borehole" data of figure 34 in the same attribute

55

space. If all goes well, in figure 42, the validation data cluster 199 in figure 42 will be positioned adjacent to the training data cluster 197 in the attribute space of figure 42. The validation data cluster 199 of figure 42 should cluster adjacent to the training data cluster 197 of figure 42 because the class of the attribute data in the "validation data" of figure 37 was assumed to be the same as the class of the attribute data in the "training data" of figure 34.

When the execution of the "training of the classifications" code 242 of figure 39 is complete, the "validation successful" block 212 of figure 38 and 39 is executed. The "validation successful" block 212 of figure 38 will have a "yes" output (indicating a successful validation) when the "validation data" cluster 199 of figure 42 clusters adjacent to the "training data" cluster 197 of figure 42. However, in figure 38, if the validation is not successful (a "no" output from the validation successful block 212 of figure 38 is generated), the operator can now edit the "input data" and start again - that is, the operator can edit: the "input attribute selection" 134, 136 and/or the "training/validation" 182, 184 and/or the "parameters" 138 and/or the "classes" 188 in figure 38. This can be done by utilizing the edit block 214 of figure 38. In the edit block 214 of figure 38, the operator can now edit the "training/validation" input data 214a, or the operator can "select/redefine classes" 214b, or the operator can "edit input attributes selection" 214c, and/or the operator can "edit parameters" 214d. After the above referenced "input data" is edited via the edit block 214 of figure 38, the "training of the classifications" block 208 of figure 39 is reexecuted in the manner discussed above. However, on the next pass, if block 212 of figure 38 is satisfied, the validation will be successful, and the "apply the trained classification determine quality indicators" code 216 of figure 38 can be executed.

In figure 43, when the "apply the trained classification..." code 216 of figure 38 is executed, all of the attribute data 201 of figure 43 on the attribute.data sets 54a, 54b of the "input attribute selection" of figure 38

56

will be used to generate a "cross plot in attribute space", such as the "cross plot in attribute space" 203 illustrated in figure 43.

In figure 44, when the "cross plot in attribute space" 203 of figure 43 is generated, that cross plot 203 is used to generate a classified result 130, such as the class data set 130 in figure 44. Recall that the individual attributes (ax, bx) on the attribute data sets 54a, 54b of figure 43 are plotted in attribute space, yielding the cross plot 203 of figure 43. Different labels, such as colors, are assigned to each cluster 203a, 203b, 203c, and 203d in the attribute space 203 of figure 43. Now, each of the individual attributes (ax, bx) have a label, such as color, unique to each attribute. In figure 43, the locations on the attribute data sets 54a, 54b of figure 43 are now provided with labels, such as the aforementioned colors, that correspond to the attribute data located thereon. The result is a class data set, such as the class data set 130 of figure 44. A sample classified result 130/class data set 130 is illustrated in figures 20 and 21.

The class data set 130 comprises a plurality of areas, where each area has a label, such as color. In the "supervised" classification case, the label (i.e. color) of an area on the class data set indicates one of the classes of figure 31. For example, using a color label as an example, a color red in a first area on the class data set 130 may indicate that the first area has the "sand" class 174a of figure 31. In the "unsupervised" classification case, the color red on a second area of the class data set indicates a particular characteristic of that area which is different from the characteristic of a third area on the class data set.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

1. Introduction

SeisClass provides methods and tools for discriminating statistically and
non-statistically between different classes based upon seismically
derived or any other attributes. Typical seismic attributes are amplitude,
instantaneous attributes, and statistical moments. Classification may
be applied to lithofacies estimation (sand, shale, etc.), pore fluid type
estimation (gas, oil, water) and stratigraphic characterization.

Traditionally attribute maps have been used one by one to reveal the
secrets of the subsurface. However, with the increasing number of
attributes in addition to well logs, it becomes an intractable problem for
the human interpreter to combine all this information into a few
lithofacies or property classes that the interpreter is really interested in.
The main goal of SeisClass is to assist interpreters classifying
subsurface phenomena into lithofacies classes using many attribute
maps and well logs.

Refer to Figure 45.

SeisClass offers a variety of classification methods including
"unsupervised" methods which attempt classification based upon
natural clustering of data in attribute space and "supervised" methods,
which require the provision of training samples.

2. Specifications

Refer to Figure 46.

SeisClass may operate in different domains:

58

¥ "Horizontal Classification", using surface attribute grids as data source and geometric reference (which may later be extended to 2D line representations). The GridMap window serves as visualization tool for the individual attributes.

5   ¥ "Vertical Classification", using a zero-offset (vertical seismic) section from a seismic pre-stack cube as geometric reference: attributes, that are going to be used in classification are represented as vertical attribute sections referring to the zero-offset geometry. The section window serves as visualization tool for the individual attributes.

10   Refer to Figure 47 relating to Data Access Vertical Classification.

Attributes are generated evaluating the amplitudes in the pre-stack cub along the travel time curves/surfaces previously constructed or picked. The amplitude variation versus offset, AVO, is approximated by a "least squares" polynom. The polynomial coefficients, the attributes used by

15   SeisClass, are geometrically associated with the zero-offset section. Any other attribute section having the same geometrical association can of course be used as additional classification input, e.g., interval velocities.

The specification generally regards the above classification domains as equivalent. A horizontal classification domain is often taken as example

20   throughout the specification in order to keep the document readable. A dedicated section, "Vertical Classification (2.10)", deals with the specific differences in that approach. The main specification will refer to this section where ever it is necessary.

The section "Classification Methods (2.2)" lists SeisClass' repository of

25   different classification methods. This section offers references to the theoretical background; some details are outlined in the "Appendix C Classification Method Reference".

The management of data involved in a classification require some extensions of the GeoModel. The new data entities are introduced in "SeisClass Data Entities 2.3" The section "Data Management (2.4)" presents the SeisClass data model from the application point of view. A

5   detailed "Event List for Database Objects (D.1)" that illustrates the use of the SeisClass data entities can be found in "Appendix D."

SeisClass introduces tow new, dedicated windows, the "Administration Window (2.5)" and the "Crossplot (2.6)". The "Administration Window (2.5)" serves as SeisClass data and process manager. The specification

10   covers dialogue layout and the functional description.

"Crossplot (2.6)" is the data visualization and exploration tool in the attribute space – "show all attributes at a given position". It offers selection and deselection capabilities of data in the attribute space. This section describes the local functionality and the human interface.

15   Depending on the classification domain, either the GridMap or the section window serve as the individual attribute visualization tool – "show all positions for a given attribute". As mentioned above, the horizontal classification is used as example, thus the section "GridMap Extension (2.7)" covers the SeisClass specific functional specification for

20   the GridMap window. For the vertical domain, see "Functional Extensions Section Window (2.10.4)."

SeisClass requires very tight communication between a crossplot and an attribute visualization window. The section "Interactions between Windows (2.8)" is dedicated to this topic from a user's point of view.

25   The generation of attributes is considered to be independent of SeisClass. In principle, attribute generators are readily available in Charisma. A smooth work flow, however, requires substantial

60

overhauling of this module. The section "Attribute Generator (2.9)" collects the specifications for this part of IMain.

The section "Vertical Classification (2.10)" covers all special features and exceptions for classification in a vertical domain. It can he seen as an extension of the functionality required for horizontal classification.

2.1 Product Perspective

SeisClass is based on the Charisma 4.0 architecture. Charisma 4.0 is the first release that tightly integrates with the GeoFrame well data. All other data types are still modeled and managed by GeoModel. GeoShare will provide the exchange of these data types. A dedicated link for girded surfaces is under consideration. The following sketches the position of SeisClass versus GeoFrame.

Refer to Figure 48.

SeisClass moves onto GeoFrame pick-a-back on Charisma IMain. GeoFrame/SDM is hidden from SeisClass by IMain's application data access layer (CDM). The GeoFrame ITC and database update notifications are implemented for well data objects following the Charisma 4.0 specification.

SeisClass can be launched from within Charisma IMain or as a separate module SeisClass. The SeisClass specific functionality is license controlled. It will not be available, i.e. insensitive if no license is available. License handling and human interface follow the InDepth example. Details can be found in "Appendix E".

When launching SeisClass, the user faces two new windows that serve SeisClass specific purposes.

Refer to Figure 49.

IMain, not all windows are equally important in a SeisClass scenario. The Administration window stands out as the main interface for running a classification and data management. The crossplot and, depending on the classification domain, the map or section window serve as data exploration windows. The other windows, positioned on the lower left, play only a secondary role and can be iconified on demand.

Refer to Figure 50 for Window Usage depending on Classification Domain.

10   2.1.1 <u>Context Diagram</u>

The methods offered in SeisClass can be grouped into:

- unsupervised methods that perform classification according to the input data's clustering, and
- supervised methods that require a manual class definition and a
15   training step, a manual cluster definition.

Therefore we present context diagrams for both method classes, beginning with the diagram for unsupervised methods because fewer data objects are involved.

Refer to Figures 23, 25, and 26.

20   A stack of "input attributes" and "optionally supplementary well attributes" are input "to the unsupervised classification process". The input attribute bulk representations – grid for horizontal; section for vertical classification – have a common, regular positioning. The attribute vectors with which the classification process is dealing are
25   generated by sampling the values in the attribute stack at a common position. (See also "Input Attributes to Attribute Vectors".) The well

supplement, however, is positioned highly irregularly, i.e. at the bore hole locations. The "Circle 1" left of the process box represents the conversion attribute vectors. After this transition, position information is no longer part of the vector; it is kept as source reference.

5   Depending on the classification method used, "parameters" can influence the classification result. Data dependent parameters may supplement purely method dependent parameters in the course of a classification process. The data dependent parameters can be seen as part of the classification result, the output binding, or as feedback

10   information to refine the classification. The feedback loop, indicated by "2" in "Unsupervised Classification Context Diagram" is not necessarily transparent to the user.

In the "classification process" the relative attribute vector location is evaluated against the overall attribute vector distribution. Following the

15   rules of the method used, one single class is assigned to the input attribute vector's location, i.e. grid or section cell. The "classified result" is saved using a geometric representation identical to that of the input attributes.

The "classified result" may be complemented by other grids or sections

20   describing the quality or probability of the classification result.

The context diagram for supervised classification can be seen as an extension of "Unsupervised Classification Context Diagram:"

Refer to Figures 30, 38, and 39

Class definition and attribute vector to class assignment is manually

25   accomplished when running a supervised classification. This is a highly interactive process. Before starting any supervised classification process, the classes that are going to be used must be defined "classes in use".

63

This a priori decision is a part of the process parameterization. It includes also class dependent parameters.

Like the "well supplement" to the "input attributes", some bore holes may also be identified as training/validation data, the "well training/validation supplement". However, the usage of "well training/validation supplement" will be limited to horizontal classification. (See "Training/Validation Data from Wells" for further explanation.)

The loop between "classes in use, training/validation" and the "supervised classification" process implies the need for high interactivity.

The following figure shows another aspect of the functional context, the sequence of actions in a classification process, both for unsupervised and supervised classification. These figures are the basis for the "Scenario Drive Events Lists."

2.1.2 <u>External Interfaces</u>

As outlined in the "Product Perspective," SeisClass will either be running in a Charisma 4.0 or in a heterogeneous GeoFrame 3.0 environment.

2.1.2.1 <u>Charisma 4.0 SeisClass</u>

Figure 51 focuses on the flow of data types between applications that are relevant for SeisClass.

A description of data types, that are exchanged in this context can be found in "Table 2: Data Types Exchanged in SeisClass/Charisma 4.0 Environment."

Prime input data are surface attributes (surface data in "Table 2:") either generated by IMain's "Attribute Generator", the "Correlation Map" function or imported attributes provided by any other external application. Usually these surface attributes are associated to an

5   interpreted horizon or layer; this interpretation is provided by IMain "Seismic Interpretation" or by "ASAP," IMain's automatic tracking module.

Qualitative and quantitative well control is provided by "Synthetics", "StratLog" and "GeoGram+". "StratLog" offers multi-well, geological

10   correlation and identifies the sequences to interpreted and classified. "GeoGram+" provides a rigorous velocity log calibration in order to build a reliable depth-time relation. Once the well logs are prepared, "Synthetics" creates the synthetic seismograms that are used to tied the seismic interpretation domain to the well interpretation domain. Then

15   the wavelet extraction function in "Synthetics" supplies the calibrated, optimal synthetic well reference.

SeisClass collects the well logs and synthetics in the well supplement to establish the quantitative well control. The synthetic seismograms are the basis for an identical attribute generation "on-the-fly" in SeisClass,

20   since the surface attributes provided the "Attribute Generator" allow an exact reconstruction of the process and parameters.

SeisClass communicates with "GeoModel" to keep track of all data bindings and parameter settings that are required in a classification run.

The output of SeisClass, the classified result and quality measures, may

25   be selected in various visualization applications. "MapPlot" may be used to provide high quality hardcopies, "GeoCube" and "GeoViz" offer 3D visualization on composite displays.

The following "Table 2:" lists all the data types, their storage environment and access type. Most important is the type "Surface Data". This term is to be understood as surface attribute representations of type grid that may have boundary (fault) information. Class Grids, the main output of SeisClass, are also "Surface Data", but are mentioned explicitly where applications have to deal with this new type of "Surface Data", namely "MapPlot," "GeoCube" and "GeoViz".

Table 2: Data Types Exchanged in SeisClass/Charisma 4.0 Environment

Applications	G oModel		GeoFrame	
	reading	*writing*	*reading*	*writing*
Imain *Seismic Interpretation*	Seismic Data Surface Data WellTies	Surface Data WellTies	Well Well Logs Well Markers	
IMain *ASAP* Horizon Tracking	Seismic Data Surface Data WellTies	Surface Data		
Charisma *Synthetics*	Seismic Data Surface Data WellTies	WellTies	Well Well Logs Well Markers	Reflectivity Synthetics (Well Logs) Well Markers
GeoFrame *StratLog*			Wells Well Markers Well Logs Surface Data[a]	Wells Well Markers Well Logs Surface Data[a]
GeoFrame *GeoGram+*			Wells Well Markers Well Logs	Calibrated Logs Reflectivity Synthetics
Imain *Attribute Generator*	Seismic Data Surface Data	Attributes (Surface Data)		
Imain *Correlation Map*	Seismic Data Surface Data	Attributes[b] (Surface Data)		
IMain/Stand-alone SeisClass	Attributes (Surface Data) WellTies	Class Grids, QC-Attributes (Surface Data)	Well Markers Well Logs Synthetics	
Charisma *MapPlot*	Seismic Data Surface Data Class Grids		Wells Well Markers[c]	
Imain *GeoCube*	Seismic Data Surface Data Class Grids		Wells Well Logs Well Markers[c]	
Charisma *GeoViz*	Seismic Data Surface Data Class Grids WellTies		Wells Well Logs Well Markers[c]	

67

a   These data have to be exchanged via GeoShare.
b   Well supplements cannot be generated from single trace synthetics to match Correlation Map results.
c   Required new functionality, independent of SeisClass.

5   ## 2.1.2.2 GeoFrame 3.0 SeisClass (IESX Scenario)

The importance of surface data has been stressed in the previous section. This requires smooth surface attribute data exchange between IESX and SeisClass. Surface attributes in grid representation are part of the GeoFrame 3.0. The following scenario assumes that GeoFrame 3.0

10 attribute grids and fault boundary descriptions are transparent and equally accessible through SeisClass.

"GeoModel" is used as classification run administration data storage. It relates the input attribute grids, the well supplement, classification parameters and classified output for each classification run.

15 Generally all components shown in "SeisClass in Charisma 4.0" are also available in this scenario: some of them, however, have been omitted for the sake of clarity. It is assumed that the seismic data are exclusively handled inside the IESX environment and not transparent to SeisClass.

IESX time interpretation is gridded and released to the grid storage in

20 "GeoFrame". The surface attributes are produced by the "Computation Manager" and put out into "GeoFrame", stored in the same grid library.

Since the seismic data are local to the IESX environment, the seismic-to-well correlation must occur in the IESX domain as well. "IESX Synthetics" is supposed to deliver the calibrated synthetic seismograms

25 to the GeoFrame well log storage. It is supposedly necessary to generate the attributes, corresponding to the "Computation Manager" results, also in "IESX Synthetics", because methods and parameters cannot yet be communicated through GeoFrame.

68

The modules "CPSLink" and "CPS 3" take the place of "MapPlot" in "SeisClass in Charisma 4.0". This requires, however, that "CPS 3" is able to handle index grid representations (class grids).

Interactive visualization is done in "GeoViz" (the Charisma 4.0 variant).

5   Refer to Figure 52.

2.2 Classification Methods

The methods that SeisClass is using are catalog controlled. New algorithms can be made available as soon as the method and method/parameter catalogs have been updated with their properties and
10   attributes. (See "C3 Tables: Classification Methods".)

Six different methods are part of the first commercial version of SeisClass. They belong to classification methods of either of the following groups:
*   "unsupervised" or
15   *   "supervised"

2.2.1 Unsupervised Methods

Unsupervised methods base their classification on the clustering of the entire dataset in attribute space. The result does not depend on a priori information.

20   The usage of unsupervised methods is appropriate where one of the following exist:
*   no reliable a-priori information is available
*   an un-biased cluster analysis should be carried out in order to verify the existence of clusters and their respective separation

69

The number of classes, i.e. the number of clusters in attribute space is the dominant parameter for the two unsupervised methods offered.

2.2.1.1 K-Means Clustering Method

K-Means Clustering is a classical algorithm due to MacQueen [M1967]. It is often used as an initialization of computationally more expensive algorithms. The algorithm will always converge, but might give different results for different runs due to local minima in the objective function used by the algorithm.

K-Means Clustering does not make any assumptions about the distribution of the data. Depending upon the number of classes "k", "k" cluster centers are randomly selected from the available attribute vectors. In the classification process, all attribute vectors are assigned to one of "k" clusters based upon minimizing the distance from the vector to the mean of the cluster. This is done by iterative improvement, thereby updating the cluster means after each new assignment. The iteration is carried on either until there are now assignments or until a maximum number of iterations has been reached. The initial positions of the cluster means are randomly chosen by selecting "k" different attribute vectors from the entire attribute vector collection.

"Advantages"
- fast convergence
- no assumptions about underlying data and data distribution
- well-proven classical method

"Disadvantages"
- results depend on initial selection of cluster centers which is usually method controlled
- method does not always select expected clusters.

Refer to Figure 19.

2.2.1.2 Competitive Learning Method

Competitive Learning is usually labeled as a neural network paradigm
and considered as an alternative approach to statistically motivated
5   methods like K-Means Clustering. However, one may as well view the
algorithm as an extended variation of K-Means Clustering. Comparative
studies with Competitive Learning have yield good results in several
fields.

Like K-Means Clustering, Competitive Learning does not make any
10   assumptions about the distribution of the data. The number of classes is
specified a priori. The method itself controls the initial cluster center
distribution. During training, attribute vectors are iteratively presented
and the cluster vector that is closest to the attribute vector is updated so
that it is even more likely to win the next time the particular attribute
15   vector is presented. Training is stopped after a certain number of
iterations, or when the cluster vector only changes marginally.

"Advantages"
- produces same results for identical initial user-supplied conditions
- no assumptions about underlying data and data distribution

20   "Disadvantages"
- method does not converge in general, but this is compensated with
  weight decay
- slow training of network
- the theoretical foundation is not as solid as classical statistical
25   methods

Refer to Figure 53.

71

2.2.1.3 Comparison of Unsupervised Methods

In general one can expect Competitive Learning to perform better, but only after very time-consuming training runs. It is shown how the methods perform on a simple dataset with four natural clusters. The

5 Competitive Learning algorithm moves slowly, but steadily toward the cluster centers while the Competitive Learning algorithm always finds the optimal solution. The K-Means algorithm will typically find something similar to that, but sometimes will also come up with the perfect solution. Note, however, that this is a highly artificial training

10 set. One may conclude that in general, Competitive Learning is expected to produce better results, but at a much higher price.

2.2.2 Supervised Methods

Very often detailed and reliable information is available at wells. This information is too important not to be considered in a classification

15 process. Therefore, it is desirable to feed in this auxiliary information.

On the other hand, there might be a clear question to be answered by a classification. The classes involved in the play are given by an interpretive analysis of the entire system rather than a statistical definition. The margins for modifications or adaptations of class

20 definitions can be very limited.

The available attributes might not be ideally suitable or fully describe the system properties that are to be classified. A priori knowledge can supplement the attribute data and compensate for missing and not fully quantifiable attributes.

25 Usually the a priori information is fed into the system as training data, i.e. attribute vectors that have a known class relationship. It is evident that a substantial number of attribute vectors have to be made available

72

for each class in order to get a proper statistic description of the data distribution. Under the assumption that sufficient data are available for each class, a subset can be taken away from the training data and be reserved as validation data. A successful classification will predict the correct class at the validation data locations.

SeisClass offers three statistical and one neural net type of supervised methods. The statistical methods will be discussed first, ordered by their level of sophistication.

2.2.2.1 Fisher Method

Fisher's discriminant function is based on the assumption that the distribution of the classes is equal meaning that the energy distribution in the covariance matrices is equal. The goal is to find projection functions (discriminant function) for each pair of classes that maximize the distance between the training data populations. An attribute vector is then classified by using the discriminant function to find the class to which it has least distance.

"Advantages"

- fastest statistical classification method
- requires only training data definition, no parameters
- any type of data distribution, as long as the distribution of the training data per class are equal

"Disadvantages"

- overlapping data distributions/bad training data obscure classification result
- the requirement of equal training data distribution (equal covariance matrix) may be difficult to achieve. In practice the method is nevertheless very robust against violations of this condition.

73

For further details, see "Section C.2.1.1".

2.2.2.2 Bayesian Method

This method derives the class probability from the training data. The implementation of the Bayesian method in SeisClass requires Gaussian distribution for all classes. According to that attribute vector's position in attribute space, the probability for belonging to a certain class is determined for all classes. The class giving highest probability is assigned to the vector. If the highest probability is lower than the required confidence level, the "doubt class" is assigned instead. The "outlier class" is assigned, if the vector is more than a predefined distance away from the most probable class center.

"Advantages"
- rigorous mapping of the individual class probabilities; good opportunities for quality assurance
- handles automatically doubt and outlier classes
- supports a prior class proportion definition

"Disadvantages"
- the requirement of Gaussian distribution and equal training data distribution is difficult to achieve. In practice the method is – like the Fisher method – robust against violations of this condition.

For further details, see "Section C.2.1.2".

2.2.2.3 Contextual Bayesian Method

In addition to an individual point-by-point approach, the Contextual Bayesian method considers additionally the neighborhood of the classified location. The configuration probability is defined for each class. Three different shapes, contextual prototypes are offered.

74

Locations that have been classified as "doubt", can be re-classified because of the class context of its neighborhood.

"Advantages"

- resulting class maps coherent, having a low noise level
5 - rigorous mapping of the individual class probabilities; good opportunities for quality assurance
- handles automatically "doubt" and "outlier" classes
- supports a priori class proportion definition

"Disadvantages"

10 - the requirement of Gaussian distribution and equal training data distribution is difficult to achieve
- method is comparatively slow

2.2.2.4 Back Error Propagation Method

The back-propagation algorithm is the most popular and widely used
15 learning method in neuro-computing today. It is easy to understand and to apply and is also very powerful. See Lippmann for an introduction, and Hecht-Nielsen for a more in-depth analysis of this learning law. It works by doing iterative gradient descent in weight space, minimizing the squared error between the desired and actual outputs.

20 Refer to Figure 54.

A single layer perceptron has the same representational power as a linear Fischer discriminant classifier. Back-propagation type networks work by constructing hyperplanes, thus dividing the n-dimensional hyperspace into decision regions. These nets often successfully compete
25 with Bayesian classifiers, especially in cases of unknown, highly non-gaussian input probability density functions.

75

"Advantages"

- can resolve non-linear attribute/class relations

- no assumptions about data distribution

"Disadvantages"

5 • method is comparatively slow

- no rigorous theoretical framework that relates the output to input
  and justifies the output from method and input data; the method
  works rather as a black box

- the random network initialization can cause different results each
10   time the method is applied; the degree of variation in the result can
  however be seen as a quality indication.

For further details, see "Section C.2.1.4".

2.3 <u>SeisClass Data Entities</u>

SeisClass introduces some new concepts into the existing data model.
15 The most striking of all is the idea of a discrete property in contrast to
continuous properties, that up to now have solely been dealt with. The
first parts, Classes (2.3.1) and Class Representations (2.3.2) are
dedicated to the introduction of this concept.

The new process "classification" requires necessarily its own
20 input/output binding and parameter management, as long as the results
are supposed to be repeatable, provable and verifiable. This follows the
concept of a "module-run" that is already established in GeoFrame and
parts of GeoModel "Classification Run (2.3.6)".

The experiences gained with the use of the prototype suggest the
25 collection of input data into sets that are likely to be re-used. The
required integration of surface and well data adds some complexity. Also

76

here, collection techniques that have already been established earlier "Extensions to general set (2.3.4)" are re-used and extended where necessary, "Well Binding (2.3.5)".

2.3.1 <u>Classes</u>

5   Attributes are currently referring to a measurement, defining the physical context of the property being quantified. The assumption is that the measured quantity describes a continuous property. Although the preferred reconstruction rules may vary from measurement to measurement, it is generally acknowledged that the data can be

10   "interpolated".

Discrete properties break with this concept. The discrete value has no meaning in itself; it must be translated into the discrete property description. This is usually implemented by a look-up-table that joins the index (discrete value) with the full property description.

15   SeisClass uses a class library as repository for class properties. (See "Class Library (2.5.9)".)

The purpose of the classification as well as the classification method require different types of classes that are normally not used in one and the same run. This suggests the use of class families.

20   The class library is supposed to be a resource open to other applications performing similar tasks. The GeoFrame RockClass application, e.g. accomplishes lithological classification on log data. Unfortunately, RockClass is not using a transparent class repository and an implementation of such as repository is not planned in the GeoFrame

25   3.0 time frame. It is nevertheless attractive to combine interpretations of RockClass with SeisClass. The level of detail, resolution and even ambition in RockClass may be substantially different to that in

SeisClass. The concept of partitioning class families into many different levels and gradations allows parallel use of the class repository from applications working in entirely different domains.

SeisClass will deliver a default class repository, a starter class library

5   that may be extended to meet specific needs when performing particular classifications. This is indicated by the global area in "Example Class Hierarchy."

Unsupervised classification methods will require their own set of classes. The default class repository provides these definitions. A class family tag

10   prevents mixing class definitions in a particular classification run across class families in a particular classification run.

The classes have attributes such as a class color. The user is provided with means to control the attribute settings on a project level through a dialog described in conjunction with Section 2.5.9 (Class Library).

15   Refer to Figure 31. Figure 31 shows an example of a class repository consisting of some class family trees. Classes from different families are never expected to be used in a single classification run.

Furthermore, some classification runs may require very data or problem specific class definitions. It is therefore unlikely that those classes are

20   being re-used in other projects. Classes may therefore be specified locally for a project. This requires, however, that project backup and restore must take care of and maintain the class definition repository.

2.3.2 Class Representations

The class definition repository or class library provides the foundation

25   for discrete surface attributes and their bulk representation. However, a direct coupling of bulk data samples to a common class repository by

78

- a method type description, defining the method to be a "discrete" or a "continuous" classification
- a supervision type description, defining whether the method is "unsupervised" (no training/validation data required) or "supervised"

5   (training/validation data mandatory)
- a descriptive string, further describing the method, i.e. for the "Back Error Propagation" method "Artificial Neural Net"

"Table 9: Classification Method Catalogue" in "Appendix C.3 Tables: Classification Methods" presents an overview over the methods. See also

10   the relations to the parameter catalog (next chapter "2.3.3.2") and the classification run entity (See "2.3.6 Classification Run".)

2.3.3.2 <u>Parameter Catalog</u>

All parameters that are used by any classification method ("2.3.3.1 Method Catalog") are registered in the parameter catalog. A parameter

15   definition is uniquely linked to a method in the method catalog. The intention of this parameter catalog is to provide the following:
- as much information as possible to build a method dependent user interface for method parameterization and
- reasonable defaults for process initialization that may later be

20   overruled by classification run specific settings

Among the parameter catalog attributes are the following:
- the parameter name that is used in the user interface when referring to this parameter
- a value type specification defining whether the parameter is an

25   integer float or vector
- an identifier denoting parameter validity, either "per method" or "per class " (having a link to the class catalog; see "2.3.1. Classes")
- an identifier denoting whether the parameter should be transparent, i.e. editable for or hidden from the user

- and finally a default value and value range description. The definition of default value and range are optional depending on the parameter and method.

"Table 10: Method – Parameter Catalogue" in "Appendix C.3 Tables:
5 Classification Methods" presents a complete list of parameters, attributes and defaults per method. See again the relational diagram.

Data dependent information can also be saved as 'hidden parameter'. This will be the case when e.g., discriminant functions have been built, using training data and method parameters; once defined, the
10 discriminants could also be applied to other similar input data sets. This includes the 'statistical signature' for statistical methods.

2.3.4 Extensions to "general_set"

The "general_set" did successfully solve the InDepth problem of collecting representations belonging to various different data model
15 domains. SeisClass faces a similar situation. The goal is to group the following:
- input attribute representations, both for horizontal and vertical classification, into re-usable sets,
- corresponding well-derived attributes to individual attribute
20 representations,
- training/validation representation and well training/validation data.

This resembles the velocity function "velocity_func" in InDepth. The multiple bindings to well data items require, however, an extension.

Refer to Figure 57.

25 The implementation requires two new sub-types of "general_set", the "attr_set" and "attr_well_set" objects. The only purpose of "attr_set" is to group a number of "attr_well_set" objects. Each "attr_well_set" object

80

refers to an attribute representation (either surface or section). "Attr_well_set" objects can be referred to by well supplements. The well supplements are implemented through "attr_well_ref" that split up into either logs or constants. (See "Section 2.3.5" for details.)

5   In order to simplify the bulk data access, the attribute representations in an attribute set must refer to a "common geometry". This means that all members must refer to the same grid library (horizontal classification domain) or the same seismic line geometry (vertical classification domain). The application is responsible for enforcing the geometry

10   constraints. Mixing grid libraries or geometries in an "attr_ set" or "attr_well_set" is a violation of the data model and is considered an error.

Analogous structure has been selected for the training/validation set to keep a symmetric data model though the training/validation set is conceptually less complex. A training/validation set – class_train_set –

15   will consist only of one discrete attribute representation. It may be accompanied by a well supplement consisting of class-to-well assignments, symmetrically to the input attribute set structure.

Refer to Figures 13, 58 and 59. The use of new sub-types of "general_set" is illustrated in the entity relation diagram in Figure 55. It

20   includes all variants of "general_set" types, namely "velocity-func".

2.3.5 Well Binding

Refer to Figure 60.

The requirement to always extract well attributes from well logs has been regarded as too strict since the existence of attribute logs for all

25   wells is necessary. It becomes even more evident for training/validation logs. The introduction of constants significantly improves the usability. Attributes can sometimes be provided more easily by user defined constants than by a comparably heavy log interpolation at the well trajectory/horizon intersection. Additionally the class assignment for a

81

given well is more conveniently done via a constant than a class log interpretation.

The flexibility with respect to usage has its price in an increased data model and HI complexity. Figure 60 shows in detail which intermediate

5   objects have to be added in order to branch from "well_attr_ref" to logs or constants having continuous or discrete representations.

2.3.6 Classification Run

This "module_run" - equivalent "class_run" – is the main administrative instrument. It keeps track of the input binding, parameters and

10   classification status report information on all data involved in a classification can be provided at any time.

As seen from the data model, a classification run can exist on its own. However, it is recommended that classification runs without any reference to other objects are removed.

15   In figure 61, the presentation of the various links to other data entities is organized according to figure 61 counter-clockwise starting from "model entity" at the top.

Fundamental Properties

The classification domain is set by a link to a "model entity". It is a

20   structural surface (grid) for horizontal classification or a seismic section for vertical classification. This link does not constrain the actual geometry for input sets which is the responsibility of the application. (See "common geometry".) Since the representation type is not restricted, the model is also open for an extension to 2D horizontal classification.

25   A classification run uniquely relates to a classification method from the method catalog "class_method_cat" which has already been illustrated. The type of method regulates the following:

- which input bindings are required
- what output can be generated and
- the parameter suite

The definitions for each method are given in "Section C.3", Table 8,
5 "Classification Methods, Overview", Table 10, "Method - Parameter Catalogue".

Input Binding

The link to one input attribute set "attr_set" is mandatory for all classification methods. (See "Section 2.3.4".) If a method is "supervised",
10 the training/validation set "class_train_set" becomes a mandatory input.

Refer to Figure 61.

Parameters

Parameters or intermediate results, i.e. grain during network training are kept in "class_run_params" objects. Method parameters not explicitly
15 represented in "class_run_params" are supplemented by the defaults defined in the classification parameter catalog "cmeth_coeff_cat". The application, i.e. the classification process, will always be able to construct a complete set of parameters. Meaningful defaults or sensible 'reset' conditions can always be provided. For details, see Table 10,
20 "Method - Parameter Catalogue".

It is important to know at any time which classes are involved in the classification process. The "class_usage" relation provides this information. The "class_usage" relation will be used when new discrete representations are to be created, e.g. training/validation sets or
25 classification results.

Output Bindings

The primary result of a classification run will be the class map or class section. Depending on the classification type, it will either be a discrete or a conventional continuous attribute representation. The classification run will keep a reference to the result if it already exists.

5   As already introduced in "Section 2.3.2", discrete representations have as many "class_decode" entries associated with them as classes that have been used "(classes_in_use)". This applies also for continuous classification results. A continuous classification result describes the transition from one class to another in form of a ratio. Each extreme,
10   zero and one, refer to a class definition via "class_decode".

Some methods can produce auxiliary output for quality control purposes. The two main types are 'confidence' and 'distance to cluster' data. Both operate on conventional representations. The classification run will keep a reference to the QC data if they have been saved.

15   Bayesian methods offer additionally the output of probabilities per each class involved in the classification. The probabilities are connected to the classification run via a relation 'probability for class_run'. The class binding is realized by "class_decode" entries which produces consistent and accurate classification results. (See above.)

20   <u>Classification Run Integrity</u>

Any changes in the parameter setting or input binding invalidate any previous results of the current, temporary classification run. Note that runs stored in the database and with persistent results are read-only and can therefore not be invalidated.

25   2.4. <u>Data Management</u>

This section discusses recommended usage of the data model extensions for SeisClass. The data model defines many relations as optional

84

because the user can and will add elements one by one. Before an input is valid in a SeisClass context, many of the objects must be in place.

The application manipulating SeisClass objects must follow the rules stated in the following sections. The user interface for data management
5   and manipulation is presented in the section "Administration Window (2.5)".

2.4.1 Input Attribute Set

The concept of input attribute sets makes it possible to share an input set in many classification runs. Modifications should be conducted in
10   such a way that they cause least disturbance.

Rule 1:   When changing the contents of an input attribute set that is already referred to by more than one classification run, a new input attribute set must be created. The existing classification runs might specifically refer to individual
15   components of the input attribute set.

Rule 2:   When changing the contents of an input attribute set that is only referred to by the current classification run, the "results_valid" flag must be set to false if any results exist.

Rule 3:   Only grids belonging to the same grid library can be selected
20   into an input attribute set (horizontal classification); only sections ("seismic_sect") belonging to the same 2D-line ("survey_line") having identical horizontal and vertical extent (CDP/time extent) can be selected into an input attribute set (vertical classification).

25   Rule 4:   On changes of the well supplement, any existing temporary results must be flagged as invalidated ("results_valid").

Rule 5:   An input attribute set can only be modified if it refers solely to the currently active classification run. Any modifications are stored in an automatically generated attribute set.

85

Rule 6:   Sets are identified by the user-defined name and the creation date. Automatic copies keep the name, but have a different creation date.

2.4.2 Training/Validation Set

5   Training and validation sets are like input attribute sets shareable between classification runs. Similar constraints as in "Section 2.4.1" apply equivalently.

Rule 7:   When changing the contents of a training/validation set that is already referred to by one or more other classification

10   runs, a new training/validation set must be created. The existing classification runs might specifically refer to individual components of the training/validation set.

Rule 8:   When changing the contents of a training/validation set that is only referred to by the current classification run, the

15   "results_valid" flag must be set to false if any results exist.

Rule 9:   Sets are identified by the user-defined name and the creation date. Automatic copies keep the name, but have a creation date.

Rule 10:   On changes of the training/validation well supplement, all

20   depending classification runs must flag the results invalidated ("results_valid").

2.4.3 Classification Run

The classification run relates input data, parameters and output data. The classification run constrains the management of objects even more,

25   as outlined by the following rules:

Rule 11:   The classification domain is a fundamental attribute. Once a classification run is created with a given domain – horizontal or vertical – it can never be modified.

86

Rule 12: When an input attribute set is linked to a classification run, the "model_entity" and the set members must belong to the same grid library or section geometry.

Rule 13: Once a classification run gets an output binding, i.e.

5   persistent database representations as results, neither the input binding nor parameters may be changed since all modifications invalidate the results.

Rule 14: Classification runs having no persistent output binding should not be instantiated in the database; temporary

10   scratch and memory representations are used in order to limit the number of persistent classification runs.

2.4.3.1 Input Binding

Rule 15: If the method defined with the classification run is supervised, input attribute set and training/validation set

15   must belong to the same grid library (horizontal domain) or "survey_line" (vertical domain).

Rule 16: Changing the input attribute set must set the run status of the temporary classification run to invalidate ("results_valid").

20   Rule 17: If the geometry of the new input attribute set is incompatible, i.e. belongs to a different grid library (horizontal domain) or "survey_line" (vertical domain), any results must be cleared.

Rule 18: "Rule 17" implies for supervised classification runs that any previous training/validation set is decoupled from the

25   classification run.

Rule 19: Changing the training/validation set for supervised classifications must set the status of the temporary classification run to invalidated ("results_valid").

2.4.3.2 Parameters, Classes in Use

Rule 20: Changing the type of classification method causes any results being cleared.

Rule 21: Any modification of classification method parameters must set the temporary run status to invalidated ("result_valid").

5 Rule 22: Adding or removing a class must set the temporary run status to invalidated ("results_valid"); parameters related to classes no longer used have to be removed synchronously.

Rule 23: Adding a class must result in a corresponding "class_decode" entry for the temporary training/validation representation.

10 Rule 24: It is not required that all classes defined in the training/validation set are actually used in a classification run. Such classes are traced upwards in the class hierarchy until a parent-class in use is found. Mismatches are signaled.

15 2.4.3.3   Output Binding

Rule 25: Each time a new classification is performed, the run status must to validated ("results_valid").

2.4.4 Hierarchical Lists

Two of the new entities will be presented in hierarchical lists: class
20 definitions and classification runs. The hierarchical lists will also require a sort of data management that is closely synchronized with the management of the database objects.

The two cases will briefly be discussed in this section, focusing on different design requirements for a generic hierarchical list management.

25 2.4.4.1 Class Hierarchy

88

Classes have a unique child-parent relationship. Each class definition can refer to one and only one parent class definition. Management of hierarchical class definition require functions to:

- to look up a parent for a given class
5 - look up the ultimate progenitor of a given class
- add a class as child of a given class
- remove a class
- re-parent a class to a given new class

The management of the hierarchy must be consistent with the data
10 model at any time. The hierarchy displays in the SeisClass administration must always reflect the database state.

2.4.4.2 Classification Run Hierarchy

Unlike the class definition, non-unique child-parent relationships may exists. This is due to the fat that input attribute sets and
15 training/validation sets can be re-used by many classification runs. In addition to the query functions in "Section 2.4.4.1", the classification run hierarchy displays in the SeisClass administration must always reflect the database state.

2.5 Administration Window

20 SeisClass handles a great number of data objects across different domains, like surfaces in seismic traveltime domain and well data in measured depth. Classification, especially supervised classification is a highly interactive and iterative process. In such an iteration process one may encounter a deadlock situation and want to step back to an earlier
25 status and follow an alternative solution.

Data selection is seen as a time consuming process. SeisClass administration provides the tools to manage selections individually and

89

re-use them in many classification runs. The previous sections have presented the individual objects in detail.

A good administration provides finally the basis for a continuous process tracking, process documentation and reporting.

5   In the following chapters, the individual dialogs involved are presented from top-level administration window down to sub-dialogs in a sequence typical for running a classification for the first time.

2.5.1 Classification Context

Refer to Figure 62.

10   SeisClass will refer any data manipulations to a current context. The most important is the classification run context which is exclusively set by the administration window.

Figure 62 shows the data objects associated to a classification run context.

15   The classification run in "Context" is exclusively valid on all IMain/SeisClass windows. Any selections and interactions relevant to SeisClass refer to this run in "Context".

A "Classification Run" can only be modified if it has no persistent output binding, i.e. the "Results" group contains only temporary (scratch)
20   representations. In order to re-execute a persistent classification run with slightly modified parameters or different input binding, a new classification run must be created ("New Classification Run") which has no output binding yet and by convenience inherits input binding and parameters from the context. The Classification Run is not permanently
25   instantiated in the database during editing and processing. Its results are temporary and are overwritten until they are explicitly saved ("Save Results Selection Dialogue").

By saving the results of a classification run, it is instantiated in the database and the result representations are permanently created. The classification run is no longer editable because the results would be invalidated.

5 During the installation or modification of a classification run's input binding, two other contexts are established, the "Input Attribute Set Context" and the "Training/Validation Set Context". These contexts are only locally valid inside the administration window and its sub-dialogs.

The context is handled strictly, implying the following rules:

10 Rule 26:   If an existing classification run is selected as active run which has associated results (existing output binding), the classification run is read-only. The read-only status is clearly visible by insensitive edit functions.

Rule 27:   An existing classification run with output binding can not be
15   modified. A new classification run must be created with the same input binding and parameters as the existing run. This new classification run will be visible in the classification run hierarchy. It is marked as temporary classification run because it is not permanently saved to the database.

20 Rule 28:   Before a request to change the classification run context is executed, the current active classification run context is tested for un-saved results. In case any results exist, the user is prompted for a decision.

Rule 29:   Changing the classification run context means that any data
25   display that depends on a previous run context is removed from the crossplot window.

Rule 30:   Selections in any sub-dialogs that have not been applied yet are lost on changing the classification run context.

Rule 31:   Changing the input binding causes the crossplot data
30   display to be cleared. See also the data management rules in "Section 2.4.3.1". The same applies for GridMap(/Section)

91

displays that specifically relate to a SeisClass classification run context.

Rule 32: Components of other classification runs can be displayed for comparison in the map or section window, e.g. training/validation data and results. The current classification run context is unchanged.

Rule 33: The contents of the SeisClass crossplot only reflect the current classification run context. Comparison of data distributions of different classification runs in attribute space is not possible.

2.5.2 Administration Main Window Dialog

The administration window serves as a data navigation and management window. The window makes use of the classification run hierarchy and uses a ListTree widget as browser. The classification context is controlled from here as well.

Refer to Figure 63.

The SeisClass Administration window consists of two main frames:
- "Classification Runs" allowing the selection of classification runs or classification run components. Here the context for all other activities can be activated or data can be selected for comparison.
- "Active Run" offers inspection and editing facilities for a given classification run in context.
  Classification runs with persistent output bindings can only be inspected using the "i" inspection button. New runs or runs without persistent results are fully editable. All required input data can be selected from this frame and classification process can be started. All interactions in other windows, such as the map or section window, will refer to this context.

The lower part of the administration window contains an optional message area and the optional one-line help area. The visibility can be controlled via the Option menu.

Refer to Figure 64.

5   The ListTree widget catches the attention of the SeisClass user with the display of the classification run hierarchy. It serves as the navigation and data selection device. It offers a hierarchical view into the classification run structures via the e.g. interpretation horizons that belong to the given classification domain (here "Horizontal Grids").

10   The hierarchical structures may be packed/unpacked by selecting the folder icons, individual components are selected by clicking on their name. The user fully controls the level of detail.

The ListTree listing contains only data belonging to a specific domain.

Classification Domain

15   Data relevant for classification are available in different representation types, for example:
• surface attributes as grid representation (default) or
• AVO attributes as vertical section representation
• surface attributes as 2D-line representations (not being implemented
20   in the first release)

Mixing attribute representations is neither meaningful nor technically convenient or possible. Therefore the a priori decision about which classification domain to focus on is useful for prohibiting the user from mixing incompatible data.

25   Changing the selection of the "Run Domain Filter" causes the classification run hierarchy to be updated.

Where ever a domain selection is required, the setting of the Run Domain Filter will be used. The risk of mixing domains is therefore avoided.

Navigation, Selections and Context Activation

5   The ListTree widget's (single) selection feature is used to identify a classification run and to select individual bulk-data representations, e.g. for "Display" in IMain's grid map window (equivalent to the "Display" function in InDepth's Administration Window). Once a classification run or dependent data are selected, the run can be activated as the
10   classification run in context by pressing the "Set Active" button.

The "Run" is determined by tracing the hierarchy of the selected item (highlighted in ListTree) upwards until a classification run object is found. The ComboBox widget, right of the "Run" label, is updated accordingly. The ComboBox itself offers an alternative classification run
15   selection on a 'run-by-run' basis that is independent of the hierarchy and its current complexity. It will offer all classification runs associated with the current reference horizon (in the case of "Horizontal, Grids domain").

Pressing "Set Active" changes the current classification context and
20   updates the Active Run area. The editability of the classification run depends upon the existence of results coupled to the classification run.

New Contexts can be generated by calling the sub-dialog for creating a new classification run ("New"...button). The selected run can be deleted ("Delete"...button); this button is only sensitive if a classification run is
25   selected.

"Run Data" reflects the current selected component. Also this is implemented as a ComboBox widget, allowing an alternative selection

mode for all data (representations) belonging to the same set (parent), e.g. all input attribute representations.

"Report..." opens a browse window that supplies descriptive information on the data selected in context.

5   Active Run

This section offers an overview over the main components of the classification run in context. The section is updated if the context is changed (e.g., by having pressed "Set Active" in the Classification Runs section).

10   Each time a classification run context is replaced by a new context, the old context is tested for any temporary results that have not yet been saved, i.e., made persistent in the data base. If temporary results exist, the "Save Results Selection Dialogue" is automatically popped up, offering the opportunity to save the results obtained in the old

15   classification run context.

Refer to Figure 65.

Each component is described by label, read-only status/selection field and can be inspected — "i" button — or modified by pressing the respective action button. The sequence reflects the logical work flow.

20   If the classification run has no persistent output binding, i.e. no results have been stored yet, the editing functions are sensitive. (See "Rule 26:" and "Rule 27:")

Not all components will be required/meaningful for all classification methods. The classification method used in the current run controls the

25   sensitivity of the component status display and the respective component inspection and edit action button.

95

- "Run Identification" – The current classification run can be modified (Modify...). The Modify...button is only sensitive if a classification run is selected.

- "Input Attribute Set" – This component is mandatory for all classification methods. The selection can be facilitated or changed by selecting an Input Attribute Set already available via the ComboBox widget. The "Select..." button calls the "Input Attribute Set Selection Dialog" which offers creation of new or manipulation of existing Input Attribute Sets.

- "Training/Validation" – This component is only required for supervised classification methods. The selection can be facilitated or changed by selecting a Training/Validation set already available via the ComboBox widget. The Select button class the "Training Validation Set Dialogue" which offers creation of new or manipulation of existing Training/Validation sets.

- "Classes in Use"

  - Unsupervised classification methods will do automatic class assignments. Only the number of classes is user-controlled. The read-only text field will signal the status: with "Modify..." the user gets the opportunity to change the number of classes being used or to define the class colors. (See also "Unsupervised Methods".)

  - Supervised classification methods require manual class assignments: the read-only text field will list the classes selected; with "Modify..." the user gets the opportunity to create/edit the current class definition. The classification method will control the actual layout of the dialog as it is also used for defining the class-dependent parameters. (See also "Supervised Methods".)

- "Method Parameters" — Some methods will require the definition of method-dependent parameters. (See Table 8, "Classification Methods, Overview". In these cases, the definition status will be shown in the read-only text; by pressing "Modify...", one may edit the method-dependent parameter settings. The respective dialog, of course, will

96

vary from method to method. (See Table 10, "Method - Parameter Catalogue"). The individual dialogs are discussed in section "Method Parameters".

- "Results"

5 A classification cannot be executed unless all required input data and parameters are set. The combination of mandatory components depends on the classification method being used. The read-only status line "Results" informs the user about whether the input specification is complete. A classification can be started or repeated.

10 - "Classify" performs the classification or a re-classification if earlier, temporary results exist. This button will only be sensitive if all required input is specified. A process completion meter and interrupt dialog will appear as long as the process is running.

In case of the unsupervised "Competitive Learning Method", the
15 neural network is trained first and the classification is started automatically afterwards.

In the case of an error that may obstruct any result or severely deteriorate the quality of the classification result, the user will be informed in advance. Error messages are presented as pop-up
20 information dialogs and outline measures for remedy.

In the case the input attribute set contains data representations that are intolerable for the method assigned to the current classification run, these representations will be ignored. The user is informed by a pop-up information dialog.

25 - "Save..." – If a classification has been performed, the current active run can be made persistent in the database, i.e. the results and the complete process history may finally be written to the database. Some methods offer optional output beyond the classification result. In such cases (see Table 8, "Classification Methods, Overview"),
30 "Save..." will lead the user to the respective selection dialog ("Save Results Selection Dialogue"). After having saved the results, the classification run entry in the hierarchical list changes its status from temporary to persistent. The action buttons become insensitive

97

(see "Edits Functions", including "Classify and Save..."); only a component inspection via the I-buttons will be accessible.

Message/Help Area

Refer to Figures 66 and 67.

5   Figure 66 illustrates the Message/Help Area. Figure 67 illustrates the New Classification Run Dialogue.

Both the message area and the one-line-help-area are optional. The Option menu will offer the respective controls.

The relative size of the message area in relation to the Classification

10   Runs hierarchy display can be controlled by a sash that is placed between the Classification Runs and the Active Run sections on the right hand side.

2.5.2.1 New Classification Run

This dialog offers the creation of a new classification run. The dialog is

15   mode-less. The dialog does not require a classification run to be selected in context. It takes, however, the current context classification run as default settings. If a classification run is selected in the "Classification Runs" list. (See "Navigation, Selections and Context Activation".) The toggle "Copy from Selection" will be sensitive.

20   A classification run is identified by a user-defined "Name" and a "Remarks" string. The classification run is uniquely coupled to a classification domain which is inherited from the domain setting in the Administration Window. (See "Classification Domain".) This setting will later also constrain the input attribute set selection. Only sets

25   consistent with the classification run's domain will be selectable. Furthermore, this dialog allows the specification of the classification "Method" and a structural "Reference", e.g. a horizon structure grid in

98

the case of a classification domain "Horizontal, Grids". If a classification run is selected ("Navigation, Selections and Context Activation"), the settings are inherited.

The toggle "Copy from Selection" offers the option to copy a classification
5 run in order to make an alternative run and compare the results. If a classification run is currently selected, this option is set by default. If no context is available, or the option is switched off, an empty classification is preset to be Fisher Discriminant. The user will create a new classification run (as a temporary classification run) by pressing "OK" or
10 abort the specification with "Cancel", returning to the Administration Window.

2.5.2.2 Modify Classification Run

Refer to Figure 68.

This dialog allows the modification of an existing classification run. It is
15 application modal: it will not be possible to change the classification context as long as this dialog is visible. The dialog is accessible from the "Run Identification" convenience button "Modify..." and from the respective "Functions" menu bar entry.

It requires a classification run to be in context as "Active Run" and offers
20 the current context as default. (See "Section 2.5.2".) The main purpose is to:

- Change the classification runs "Name" and "Remarks". These modifications do not have any effect on results eventually existing.
- Modify the classification "Method". This option is only sensitive if the
25   classification run in context has no persistent output binding. If temporary results exist, the current run will be invalidated, i.e. cleared, if necessary.

The lower section "Modify Information" of this dialog offers a preview of what may happen to the selections/settings of the current context if

"OK" is selected. Modification of these selections are handled via the section "Active Run".

"OK" will execute the modifications and return to the Administration Window. "Rest" will set the dialog state back to the current context.

5   "Cancel" will close the dialog and not apply any changes.

2.5.2.3 Delete Classification Run

The scenario outlined in Appendix E of the SeisClass Requirements Document ["SC Req"] indicates that the number of classification runs quickly accumulates. Therefore, a convenient clean-up functionality is

10 provided through this dialog.

Refer to Figure 69.

The above specification presents the user with a classification run selector (ComboBox widget in section "Run to be deleted") that inherits the context from the current classification run.

15 In the subsequent section "Delete Options," it may be specified how the components should be handled when the run in context is being deleted.
- "Input Attribute Set"
- "Training/Validation Set"

   The data model defines the Input Attribute Sets and

20   Training/Validation Sets as independent and sharable between different classification runs. The user can therefore only express his/her intention to delete the sets in conjunction with the classification run (option "Delete if not referenced"). Any references to other existing classification runs must prevent the set deletion.

25 The default setting for the delete option of the sets is "Keep the Set".
- "Classification Output" Keeping the results of a deleted classification run will generally not make sense. The deletion option default will

100

therefore be "Delete" if not referenced. The user may exceptionally over steer this option to "Keep the Results".

Class collections and parameters are always deleted as they only refer to one classification run.

5   When leaving the dialog with "OK," the user is prompted for confirmation. Upon confirmation the data will be deleted in the database and the classification run hierarchy will be updated accordingly.

"Cancel" aborts the dialog and returns the user to the administration window.

10   2.5.2.4 <u>Menu-Bar</u>

As the SeisClass Administration Window is a main window, it owns a menu bar. Next to the mandatory "File" and "Help" entries the main functions are structured into the following:

- "Functions" offering the available functions that may be executed
15   - "Options" offering options for operational behavior and window layout
- "Selections" offering the selection editors are necessary to define a classification run's components

<u>Functions</u>

The Function pull-down menu consists of three sections:

20   - The classification run handling section allow the user to create, modify, delete and save classification runs. All functions require their own sub-dialogs, "New Classification Run," "Modify Classification Run," and "Delete Classification Run." Please note that the "Modify Classification Run" dialog may be partially read-only for classification
25   runs with persistent output binding.
- "Data Analysis" leads the user to another pull-down, offering various data analysis options. "Classify" executes a classification run and "Save Classification Run..." ("Save Results Selection Dialog") finally

101

saves the classification run and its results as persistent database
objects

- The report and message window hanging section offers management
functions for "Report generation" (see "Report" for details) and the

5 message window – "Display Message Area Log" - namely full page
listings, print. The current contents of the message window can be
deleted with the "Clear Message Window" option

All functions are also available via convenience buttons in the
administration main window except the message window handling.

10 Refer to Figure 70.

Functions Analysis

- All available functionality regarding systematic data analysis before
and during classification is covered in this pull-down menu. The
contents are regarded as auxiliary functionality and therefore not in

15 the top-menu layer. However, it has a tear-off mode so that it can be
kept visible on demand.

Note: This specification is based upon the requirements document.
The availability of the functions mentioned here depends upon
further research and feasibility studies. It is a question of

20 prioritization and resources which functions are actually being
implemented.

Most analysis methods are based on statistics; their significance to
no-statistical classification methods is limited. Training Data
Validation (Hold Out Procedure) applies only to supervised

25 classification methods.

Functions that are not available/meaningful in the current
classification run context will be insensitive; the respective sub-

dialogs are automatically unmanaged if the context changes and makes the function unavailable/meaningless.

Options

Refer to Figure 71.

5 The Options pull-down menu consists of:

- Class catalog management ("Class Library...") which leads to a separate dialog (See section "Class Library".)
- Dialog layout functions allowing visibility control of optional dialog components, namely "Message Area" and "One Line Help Area". The
10 default is "Message Area" and "One Line Help Area" being visible.

Selections

The "Selections" pull-down menu allows access to all dialogs that perform selections relevant to the classification run context. Most of the selections can also be made via the convenience buttons in the section
15 "Active Run".

Refer to Figure 72.

- "Classification Domain" Selection
- "Input Attribute Set" Selection
- "Training/Validation Set" Selection
20 • "Method Parameters" Selection
- "Classes in Use" Selection
- "Save Results" Selection

The individual selection options are sensitive/insensitive depending on the classification method context.

25 2.5.3 Input Attribute Set Selection

103

The Input Attribute Set Selection dialog allows the definition of a collection of attributes to be used in conjunction with a classification. Usually the number of available attributes is limited; building collections that are re-usable in many classification runs is therefore a convenient

5   acceleration in the working sequence.

All set management functions, creation, modification and deletion are available from this dialog.

Refer to Figure 73.

Entry Points

10   The dialog is made available from the Administration Window menu-bar "Selections", the convenience Select... and the i-button in the "Active Run" area. When initiated from the i-button, the dialog is operated in informational, read-only mode.

The dialog inherits the Administration Window's "Run Domain Filter"

15   setting as pre-selection context. Only Input Attribute Sets belonging to the same domain can be created, edited or deleted.

Dialog Components

As indicated by the frame titles, the dialog contains two main sections:

- the "Input Attribute Set" – allows the set selection and set

20   management, name creation (New...)

- The "Attribute Selection" – serves as attribute selector and attribute selection manager. From here it is also possible to call the "Well Supplement Selection Dialogue"

Input Attribute Set

25   The current context can be modified via the ComboBox widget, listing all the available existing input attribute sets, sorted by reference horizon. If

104

any input attribute is already defined through a classification run context, this set is pre-selected as 'set in context'.

The "Delete" option will check whether the set in context is referred from another classification run and ask for confirmation if this is not the case.

5   After confirmation, the set is deleted in the database. Sets that are used by other classification runs cannot be deleted; the user is informed via an information dialog and a message on the message window.

Refer to Figure 74.

Attribute Selection

10   The first line shows the current "Domain" in context. Only attribute representations corresponding to this domain are selectable. Changes to the domain can only be performed in the administration window ("Classification Domain").

The second line consists of a ComboBox widget to set the "Reference". In

15   the case of domain "Horizontal Grids", this will be the list of available structural surface grids defining a reference surface and grid library. The contents of the attribute/representation settings in the lists below are filtered by the "Reference" selector setting. It will definitely be possible to select representations belonging to another surface. The grid library

20   must, however, be the same for all members of a set. The normal usage scenario assumes that it is most likely that all attribute representation belong to the same structural reference.

The left list, "Attribute Types", offers all attributes (surface attributes for horizontal classification), for which representations can be found. The

25   right list, "Attribute Representations", offers all representations for the given reference (i.e. horizon) and attribute type (i.e. surface attribute).

The lists "Attribute Types" and "Attribute Representations" are tightly connected to the "Attributes in Set" listed below. The connection is managed by the action buttons:

- "Add Attribute" – to add the selected attribute representation the Attributes in Set list
- "Remove Attribute" – to remove the selected representation in Attributes in Set from that list and
- "Remove All" – to clear all selections in Attributes in Set

Refer to Figure 75.

Any selection in "Attribute in Set" updates the following:

- Reference ComboBox widget,
- Attribute Types list and
- Attribute Representations list

Convenience Selection Behavior

The following acceleration functions will be available via double-clicks. A double-click

- in "Attribute Types" causes the first representation in "Attribute Representations" to be added to the "Attributes in Set" (provided it was not selected before).

Well Supplement

An attribute set may have an additional collection of well data, the "Well Supplement". For each well that is a member in the well supplement, a list of attributes equivalent to the attributes in Attributes in Set has to be provided. The attributes can either be directly derived from logs or from attribute generation from synthetics.

As this attribute collection is directly dependent on the Attributes in Set selection, it is invalidated if a new attribute representation is added to

106

the set. The status of the Well Supplement is shown in the read-only field between the label and the "Select..." action button that pops up the "Well Supplement Selection Dialogue". The following states are possible:

- "Supplement defined" if a well supplement selection is defined

5   (attributes for at least one well/bore-hole) and complete,

- "Supplement incomplete" if a supplement selection exists which is incomplete, i.e. some attributes for some wells are undefined or

"None" if no wells have been selected into the well supplement

Dialogue Action Buttons

10   The standard dialog action buttons perform the following functions:

- "OK" applies the current selection to the input attribute set in context according to the rules. The set selected in context is communicated to the "Active Run" which is updated accordingly if necessary.

15   Any geometry incompatibility between the set and the "model_entity" of the current classification run is signaled:

"the structural reference of the classification run is incompatible with the selected input attribute set. Adjust the classification run's reference? [Yes][No]"

20   - "Apply", the default button, same as "OK". The dialog, however, is kept on the screen. "Apply" stays insensitive if the dialog is in the read-only mode.

- "Rest" restores the selections as they were after the last "Apply". The dialog is kept. "Reset" is insensitive if the dialog is in read-only mode.

25   - "Cancel" quits any set modification and un-maps the dialog

- "Help" calls the 'help in context' for input attribute selection

2.5.3.1 New Attribute Set Dialog

Refer to Figure 76.

107

This dialog enables the creation of new input attribute sets. The classification run domain is inherited from the administration window ("Classification Domain").

As input attribute set is identified by a user-defined "Name" and a user
5   defined "Remarks" string. Both can only be set once in this dialog. It is not required that set names have unique names. In case of multiple sets with same name, the user will have to distinguish the sets by means of the creation date and time.

A set will be coupled to a structural reference, i.e. a structural grid
10   surface in case of domain "Horizontal, Grids". This link is used for sorting and defaulting purposes. The reference is set by means of a ComboBox widget.

If an input attribute set was selected in context when activation the "New..." input attribute set creation function, the selection can be copied
15   from that context. This includes all attribute selections and any available well supplements. If no set was selected in context before, the option is insensitive. Default is "Copy from current Context" on.

"OK" commits the requested set creation and removes the dialog. The new attribute set is automatically set into context of the Input Attribute
20   Set selection dialog. The user is ready to select new attributes or edit the inherited selection.

"Cancel" quits the set creation process and removes this dialog. The context of the Input Attribute Set selection dialog stays unchanged.

2.5.4 Well Supplement Selection Dialog

25   The selection of "Well Supplement Selection Dialog" manages the quantitative well control of an attribute set. The Well Supplement Selection is part of the attribute set. Quantitative well control means to

108

supplement the attribute vectors that are already given by the attribute grids with attribute vectors representing well data.

Please refer to section "Extensions to general_set" for the data modeling and specifically the sketch "Grouping of SeisClass Input Attributes" for the well binding.

The well attributes are derived from one of the following:

- well logs (at the respective horizon/grid intersection or the respective horizon's well tie)

- synthetics transformed into the 'seismic' attribute according to the history information (at the respective horizon/grid intersection or the respective horizon's well tie)

- a constant if an attribute value can be estimated but not derived from log data.

Entry Point

This dialog is only callable from the Input Attribute Set Selection dialog, "Well Supplement Select..." ("Well Supplement").

Refer to Figure 77.

Dialog Components

The dialog has two main frames:

- The "well Supplement to Attribute Set Selection" containing a table listing all wells of the well supplement versus the attributes defined by the Input Attribute Set Selection: a newly created attribute set will have an empty Well Supplement, thus an empty table will be presented.

- The "Cell Context Selection" shows the table cell context and allows its modification.

109

Both frames have their set of action buttons.

Well Supplement to Attribute Set Selection

The table catches the attention. The visible part of the table can be increased by resizing the dialog. Each well supplement attribute vector is represented by a table row. The columns refer to the attribute representations that have been defined in the input attribute selection. The column headers identify the attribute representations.

By clicking in a table cell, a table cell is activated. The "Cell Context Selection" frame is immediately updated; the well/bore-hole – identified by the row – is set as 'well in context' (ComboBox widget), the attribute – identified by the column – is set as 'attribute in context' (ComboBox widget).

The contents of the well supplement can be managed by means of the action buttons that are situated below the table. The user may add or remove table rows, i.e. add or remove a well from the well supplement by the following functions, available through action buttons:

- "By Name..." – add a well/bore hole by name, presenting a well-by-name selection dialog, a simple standard well section dialog.
- "By Well Log..." – add one or many wells that have at least one version of a named well log (channel name) in common; the selection is handled by a separate dialog, the dialog is a generalized version of the selector used in InDepth, Well Velocity Selection.
- "Remove Selected" – removes the well/bore hole identified by the current row
- "Remove All" – clears the whole table; the user is prompted for confirmation before this action is committed.

When new rows, i.e. wells/bore-holes are added to the table, its cells are filled with the text "<empty>" to mark them as 'to be defined'. Actually, any "<empty>" cell invalidates a whole attribute vector as all attributes

110

must be defined in order to be used as a complete attribute vector. Some convenience functions ("Find Empty Cells..." and "Update Empty Cells...") located under "Cell Context Selection" are offered to specifically address "<empty>" table cells.

5   Refer to Figure 78 for selection by log type.

Cell Context Selection

Refer to Figure 79 for cell context selection frame.

All information shown in this frame refer to the table cell which has been selected in the frame above, Well Supplement to Attribute Set Selection.

10   The cell context is indicated by two ComboBox widgets, identifying the well/bore-hole in context and attribute in context. These selections can be changed, thus offering an alternative way of navigation through the well supplement table.

The cell context selection can be manipulated by either of the following
15   ways:

- via cell selection in the well supplement table; updating the data in the Cell Context Selection,
- via the ComboBox widgets changing the row or column and updating the table selection.

20   The sub-frames "Generation Mode" and "Well Logs available" give a detailed description of how the attribute value is going to be generated. In total, there are three different modes available:

- "Pick from Well Log" – This option is sensitive if at least one log (having the measurement given by the attribute) for the given well-
25   bore-hole exists. The log list "Well Logs available" to the right, then shows this sub-set of well logs.
- "Generate from Synthetics" – This option is sensitive in the following situations:

111

- if at least one synthetics seismogram is available for this well/bore-hole and

- if the attribute's history information is sufficient to generate the attribute from synthetic seismic.

5   The log list Well Logs available to the right is then showing all well logs of type 'synthetics'.

- "Constant Value" – This option is always sensitive. The text-field below the option is only sensitive if the option Constant Value is selected.

10   <u>Cell Context Selection Action Buttons</u>

The following action buttons offer a variety of options for how to populate the well supplement table. The well supplement status in the Input Attribute Set Selection dialog is updated accordingly, i.e. it is set to:
- "Supplement defined" if the selection is complete
15 - "Supplement incomplete" if the selection in incomplete, i.e. cells flagged with <empty> exist, or
- "None" if no wells have been selected into the well supplement.

In detail, the following actions are possible:
- "Update Cell" applies the current setting in the "Cell Context
20   Selection" to the well supplement table.

Refer to Figure 80 for "Find Empty Cells".

- "Find Empty Cells..." is only sensitive if table cells flagged with <empty> exist. Then a simple list is offered, listing all empty cells by well/attribute reference. Clicking onto a list element causes the well
25   supplement table context and the "Cell Context Selection" to be updated. If any previously empty cell is updated, it will be removed from the list. If the last list member has been removed, the dialog is automatically un-mapped.

112

- "Update Column" – This option will be insensitive if the constant attribute value option has been selected. In all other cases, the system will try to fill all cells in the column (wells/bore-holes) with the current "Cell Context Selection". The specified well log name (channel name) is searched for each well/bore-hole and the latest version is selection into the well supplement table. Wells for which the given log does not exist are flagged as <empty>. The following function is more selective:

- "Update Empty Cells" – This function will be insensitive if the constant attribute value option has been selected. In all other cases, the system will follow the same procedure as "Update Column", but will only set <empty> cells. If this option is repeatedly executed, a specific hierarchy of well logs can be selected.

Convenience Selection Behavior

Any well attribute selection is not immediately reflected in the well supplement table above. A double-click selection in the "Well Logs available" list will cause an implicit "Update Cell" action.

Dialog Action Buttons

The well supplement dialog does not keep its own buffer of last changes. The well supplement is part of the attribute set selection which keeps a status to which it can be reset. Therefore, only a "Close" button is offered with the well supplement selection dialog:

- "Close" un-maps the well supplement selection dialog

- "Help" calls the 'help in context' for well supplement selection

Refer to Figure 82.

2.5.5 Training Validation Set Dialog

113

The Training/Validation Set selection dialog allows the definition of a set of data to be used in conjunction with the training and/or validation step during a supervised classification. Training and validation data represent a priori information and are naturally quite limited in the

5   number. Furthermore, the distribution of the a priori data is invariable within a project so that it is very likely that training data can repeatedly be used. Therefore the definition of a training/validation set is a convenient acceleration in the working sequence.

Refer to Figure 81.

10   <u>Entry Points</u>

The dialog is made available from the menu-bar "Selections" pull-down, the convenience "Select..." and through the i-button in the "Active Run" area. It is available only in a classification run context using a supervised classification method. It is read-only if initiated from the i-

15   button.

The dialog inherits the Run Domain filter ("Classification Domain") setting as the pre-selection context. Only Training/Validation sets belonging to the administration window's domain can be created, edited or deleted.

20   <u>Dialog Components</u>

The dialog comprises two main frames:
- The "Current Training/Validation Set" section enables set management (via the function buttons "New..." (see "New Training/Validation Set Dialogue") and "Delete") and gives an

25   statistical overview on the training and validation data defined in the bulk representation of the set.

- The "Training/Validation Well Selection" offers the opportunity to add well data as training/validation data.

Current Training/Validation Set

The main component of a training/validation set is the 'mask-grid', a
5   discrete index representation assigning specific cells as a training or validation location for a specific class (the type is encoded as sign-bit; as the grid is single valued, the cell can either be a training or validation data point). The geometry of this bulk data representation must be identical to the input attribute set. The data model does not provide an
10   explicit link between input attribute set and training/validation set. The requirement that training/validation set and input attribute set must have identical geometry must be ensured by the application that couples the set to a given classification run. This consistency check has to be performed if a training/validation set selection is committed ("OK" or
15   "Apply").

As existing training/validation set can be selected into context by means of the ComboBox widget, offering all training/validation sets for the classification domain in context.

Right of the ComboBox widget, the "Delete" function offers the
20   training/validation set in context to be deleted from the database. A Training/validation set can only be deleted if it is not referred to by any other classification run than the run in context. If a set cannot be deleted this is reported by an information dialog, it is ready for deletion a confirmation dialog will protect the set against unwanted deletions.

25   Right of the Delete button the "New..." function pops up a separate set definition dialog ("New Training/Validation Set Dialog"). Below the set management controls a brief statistic is given about how many training and validation samples were defined for each class. The class selection is inherited from the classification run in context. It is given by the index-

to-class-id decode relation, attached to the training/validation bulk representation.

The contents of the training/validation bulk representation are managed either from the IMain Map window or the "nD-Crossplot" window.

5   ## Training/Validation Well Selection

This frame concentrates all functionality related to the 'Well Supplement' of a training/validation set.

Refer to Figure 83.

Again the database does not have explicit links from the
10   training/validation set to the input attribute set; the individual classification run to which they are assigned provides the technical link between the sets. This means that a well can only be used quantitatively if:

- the given well is member of both the training/validation and the
15   input attribute set's well supplement,

- all attributes can successfully be generated (the well supplement table has no empty cells (See "Well Supplement to Attribute Set Selection") and

- a class had been associated with the given well which can be done in
20   this dialog section

The top part of the frame lists all wells that have been selected as part of the set. Each list entry consists of the well/bore-hole name, the type-training or validation and the source of the class definition: constant, manual class assignment or class-log (the class is read tat the reference
25   horizons log intersection). Note: Whether this option is implemented depends upon the availability, accessibility of class-logs and a consistent, common class definition.

116

The well list has extended selection policy. The multiple, extended selection feature is used to update optionally many well selections in one go. The same applies for the well removal function. The Current Well is updated on unique selection, but kept on non-unique selections.

5   A row of action buttons provides management functions for the wells in the set:

Selection of wells:

- "By Name..." – add a well-bore-hole by name, presenting a well-by-name selection dialog, a simple standard well selection dialog

10   - "By Well Log..." – add one or many wells that have at least one version of named well log (channel name) in common; the selection is handled by a separate dialog, the dialog is a generalized version of the selector used in InDepth, Well Velocity Selection.

- "Wells List from Input Attribute Set" – replaces the current list by the
15   list of wells in the current classification run's input attribute set (Well Supplement).

And de-selection of wells:

- "Remove Selected" – removes the well/bore-hole currently selected in the main list. If more than one well were selected, the user is
20   prompted for confirmation before this action is committed.

Refer to Figure 84.

The ComboBox labeled with "Current Well" allows an alternative well selection. This widget is simultaneously used to indicate the well in context because the main selection list may have been scrolled so that
25   the current well is invisible. Selection actions in the main list are immediately reflected in the ComboBox widget.

The sub-frames Status and Class Definition finally provide the class and data type.

117

Status

The options "Training" and "Validation" are always sensitive; the Status selection behavior is one-of-two.

Class Definition

5 If class-logs are available, the "from Log" option is sensitive and the selection behavior is one-of-two. The ComboBox log-selector offers all available candidates for the current well. If no class logs are available, the from Log option is insensitive and the "Constant" automatically preselected.

10 All options for "Class Definition" are presented in ComboBox widgets. The contents of the class selector is given by the classes in use class list that is a property of the classification run in context.

Well-in-Context Selection Action Buttons

Once a well selection is in context, this selection may be applied either of
15 following:
- to the well currently selected, "Update Current Well" or
- to all wells selected in the main list by "Update Selected Wells". This function will try to distribute the status and the class definition to the wells selected in the main list.

20 ## Dialog Action Buttons

The standard dialog action buttons perform the following functions:
- "OK" – applies the current selection the training/validation set in context. The set selected in context is communicated to the 'run in context' which is updated accordingly if necessary. The dialog is un-
25 mapped. "OK" is insensitive in read-only mode.
- "Apply" (Default Button) – same as "OK"; the dialog, however, is kept. "Apply" is insensitive in read-only mode.

118

- "Rest" – restores the selections as they were after the last "Apply". The dialog is kept. "Reset" is insensitive in read-only mode.
- "Cancel" – quits any set modification and un-maps the dialog.
- "Help" – calls the 'help in context' for training/validation set

5   selection.

2.5.5.1 New Training/Validation Set Dialog

This dialog enables the creation of new training/validation sets. The classification run domain is inherited from the administration window ("Classification Domain").

10   A training/validation set is identified by a user-defined name and a user defined remark string. Both can only be set once in this dialog. It is not required that set names have unique names. In case of multiple sets with same name, the user will have to distinguish the sets by means of the creation date.

15   Refer to Figure 85.

A set will be coupled to a reference, i.e. a structural grid surface in the case of domain "Horizontal, Grids". This link is used for sorting and defaulting purposes. The reference is set by means of a ComboBox widget.

20   If a training/validation set was selected in context when activating the "New..." training/validation set creation function, the selection can be copied from that context. This includes the bulk representation and any available well training/validation supplements. If no set was selected in context before, the option is insensitive. Default is "Copy from current
25   Context" on.

"OK" commits the requested set creation and removes the dialog. The new attribute set is automatically set into context of the Input Attribute

Set selection dialog. The user is ready to select another bulk-representation, define well data or edit the inherited selection.

"Cancel" quits the set creation process and removes this dialog. The context of the Training/Validation Set selection dialog stays unchanged.

5   2.5.6 Classes in Use Dialogs

This dialog may appear in two basically different variants depending upon the availability or not of class-dependent parameters.

- Class versus parameter table or
- class list

10   Above the table or list, the classification run in context is presented as a reference in a read-only text field.

The dialogs directly depend upon the classification methods. If the classification method is changed while the Classes in Use dialog is active, it is automatically un-manager; any non-applied changes are lost

15   (no warning).

Entry Points

The dialog is made available from the Administration Window menu-bar "Selections", the convenience "Modify..." and the i-button next to it in the "Active Run" area. The dialog is read-only when activated from the i-

20   button or fully editable when it has been initiated from "Select..." or from "Selections - Classes in Use".

2.5.6.1 Supervised Methods

Refer to Figure 86 for classes in use dialogs (supervised methods).

Below the table or list, the class selection and definitions can be

25   manipulated. The following action buttons are available:

120

handled by a separate dialog, the dialog is a generalized version of the selector used in InDepth, Well Velocity Selection.

- "Wells List from Input Attribute Set" – replaces the current list by the list of wells in the current classification run's input attribute set (Well Supplement).

And de-selection of wells:

- "Remove Selected" – removes the well/bore-hole currently selected in the main list. If more than one well were selected, the user is prompted for confirmation before this action is committed.

Refer to Figure 84.

The ComboBox labeled with "Current Well" allows an alternative well selection. This widget is simultaneously used to indicate the well in context because the main selection list may have been scrolled so that the current well is invisible. Selection actions in the main list are immediately reflected in the ComboBox widget.

The sub-frames Status and Class Definition finally provide the class and data type.

Status

The options "Training" and "Validation" are always sensitive; the Status selection behavior is one-of-two.

Class Definition

If class-logs are available, the "from Log" option is sensitive and the selection behavior is one-of-two. The ComboBox log-selector offers all available candidates for the current well. If no class logs are available,

the from Log option is insensitive and the "Constant" automatically preselected.

All options for "Class Definition" are presented in ComboBox widgets. The contents of the class selector is given by the classes in use class list that is a property of the classification run in context.

Well-in-Context Selection Action Buttons

Once a well selection is in context, this selection may be applied either of following:

- to the well currently selected, "Update Current Well" or
- to all wells selected in the main list by "Update Selected Wells". This function will try to distribute the status and the class definition to the wells selected in the main list.

Dialog Action Buttons

The standard dialog action buttons perform the following functions:

- "OK" – applies the current selection the training/validation set in context. The set selected in context is communicated to the 'run in context' which is updated accordingly if necessary. The dialog is un-mapped. "OK" is insensitive in read-only mode.
- "Apply" (Default Button) – same as "OK"; the dialog, however, is kept. "Apply" is insensitive in read-only mode.
- "Rest" – restores the selections as they were after the last "Apply". The dialog is kept. "Reset" is insensitive in read-only mode.
- "Cancel" – quits any set modification and un-maps the dialog.
- "Help" – calls the 'help in context' for training/validation set selection.

2.5.5.1 New Training/Validation Set Dialog

122

This dialog enables the creation of new training/validation sets. The classification run domain is inherited from the administration window ("Classification Domain").

A training/validation set is identified by a user-defined name and a user defined remark string. Both can only be set once in this dialog. It is not required that set names have unique names. In case of multiple sets with same name, the user will have to distinguish the sets by means of the creation date.

Refer to Figure 85.

A set will be coupled to a reference, i.e. a structural grid surface in the case of domain "Horizontal, Grids". This link is used for sorting and defaulting purposes. The reference is set by means of a ComboBox widget.

If a training/validation set was selected in context when activating the "New..." training/validation set creation function, the selection can be copied from that context. This includes the bulk representation and any available well training/validation supplements. If no set was selected in context before, the option is insensitive. Default is "Copy from current Context" on.

"OK" commits the requested set creation and removes the dialog. The new attribute set is automatically set into context of the Input Attribute Set selection dialog. The user is ready to select another bulk-representation, define well data or edit the inherited selection.

"Cancel" quits the set creation process and removes this dialog. The context of the Training/Validation Set selection dialog stays unchanged.

2.5.6 Classes in Use Dialogs

This dialog may appear in two basically different variants depending upon the availability or not of class-dependent parameters.

- Class versus parameter table or
- class list

5   Above the table or list, the classification run in context is presented as a reference in a read-only text field.

The dialogs directly depend upon the classification methods. If the classification method is changed while the Classes in Use dialog is active, it is automatically un-manager; any non-applied changes are lost

10   (no warning).

Entry Points

The dialog is made available from the Administration Window menu-bar "Selections", the convenience "Modify..." and the i-button next to it in the "Active Run" area. The dialog is read-only when activated from the i-

15   button or fully editable when it has been initiated from "Select..." or from "Selections - Classes in Use".

2.5.6.1 Supervised Methods

Refer to Figure 86 for classes in use dialogs (supervised methods).

Below the table or list, the class selection and definitions can be

20   manipulated. The following action buttons are available:

- "Add Class..." adds a new class to the selection 'Classes in Use'. The selection is done via a separate class selection dialog (not sensitive in read-only mode).
- "Class Properties..." is only sensitive if a class is selected in context,

25   i.e. a class is highlighted in the list or a table cells is highlighted. The

class library dialog is popped up having the current class pre-selected into context.

- "Remove Class" is only sensitive if a class is selected in context, i.e. a class is highlighted in the list or a table cell is highlighted. When activated, the class is removed from the 'Classes in Use' selection (not sensitive in read-only mode).

2.5.6.2 Unsupervised Methods

An equivalent selection is implemented for unsupervised methods. The following action buttons are available:

- "Add Class..." increments the number of classes in the current classification run. This is done by assigning all children of the class family ("n + 1") to the classification run. The maximum number of classes is limited through the highest n-classes family, defined in the class catalog (not sensitive in read-only mode).
- "Class Properties..." is only sensitive if a class is selected in context, i.e. a class is highlighted in the list or a table cell is highlighted. The class library dialog is popped up having the current class pre-selected into context.
- "Remove Class" is only sensitive if the number of classes is greater than two. "Remove Class" decrements the number of classes in the current classification run. This is done by assigning all children of the class family ("n - 1") to the classification run (not sensitive in read-only mode).

Refer to Figure 87.

Context and Standard Dialog Action Buttons

The Classes in Use dialog depends upon the current classification run context. Changing the classification run context causes the Classes in

125

Use dialog to be immediately refreshed. Any modifications that have not bee applied yet are lost.

If the dialog is in editable mode, all modifications are cached until "OK" or "Apply" are activated. The cached modifications can be un-done by selecting "Reset". "Cancel" will not apply any changes and remove the dialog from the screen. "Help" offers help in context.

"Cancel" and "Help" are the only sensitive action buttons in read-only mode.

Removing a class previously in use may have some consequences and requires clean-up actions in order to keep consistency. If auxiliary results exist that refer to the class no longer in use, the following confirmation dialog will be presented:

> "Auxiliary results referring to the removed class XXX exist. The temporary data are deleted simultaneously. Continue? Yes/No."

2.5.7 <u>Method Parameters</u>

Only classification methods that have user-controlled parameters will have their version of the Method Parameter Dialog. The method/parameter table gives detailed information. (See "Classification Methods", Overview)

Since this dialog family directly depends upon the type of method, changes of the method cause the Method Parameter dialog to be un-manager. Any non-applied changes are lost (no warning).

<u>Entry Points</u>

The method dependent dialog is made available from the Administration Window menu-bar "Selections - Method Parameters...", the convenience

126

"Modify…" and the i-button next to it in the "Active Run" area. The dialog is read-only when activated from the i-button or fully editable when it has been initiated from "Select…" or from "Selections - Method Parameters…" Note that all parameter controls will be read-only when the dialog is in informational, read-only mode.

5

The choice of controls for the individual parameters depends in general upon the method/parameter catalog definition. Depending upon the value type, see "Method - Parameter Catalogue (Table 10:)".

- Boolean parameters will be set by toggle buttons.
10 - Float parameters will be set by
  - a scale and a text-field to the right, if minimum or maximum or both are defined for this parameter [partially limited range can be extended by entering a value outside the range. If the range is fully limited, the range cannot be extended; values outside the
15 range will not be accepted.]
  - only a text field, if no range limits are defined.
- Integer parameters will be set by:
  - an editable ComboBox widget (resource XmNeditable=True), offering the discrete values as defined by the range. The range
20 may be extended by typing a new value into the editable ComboBox text field.
  - Or an editable text field only if the range is not limited at all.

Each method will be represented by its own method-parameter specification dialog. The dialog title will clearly show to which method
25 the parameter set belongs. The controls will follow the rules stated above.

Each dialog will have the standard action buttons below the parameter controls:

127

- "OK" - will apply the modifications (if any) and remove the dialog (not sensitive if read-only mode),

- "Apply" – will apply the modifications (if any) and keep the dialog (not sensitive if read-only mode),

5 - "Reset" - will restore the parameter settings to the last saved state (not sensitive if read-only mode),

- "Cancel" - will only remove the dialog and not change any settings,

- "Help" – will activate the help in context for the given method and dialog.

10 Please note that "OK" and "Apply" invalidate any existing results of the current classification run. (See "Rule 22:")

The "Status" field in frame "Active Run" will be updated. The classification must be restarted by means of "Classify" in order to restore consistent and updated results.

15 2.5.7.1 <u>Bayesian Parameters</u>

The following figure shows the parameter dialog for the Bayesian and Contextual Bayesian classification method. Both methods share the same method dependent parameters, thus they share the dialog.

Refer to Figure 88.

20 Once a classification result has been generated, changes of the Bayesian parameters are immediately applied to the class map/section.

- "Confidence Limit"
  Range: [0, 100]%
  Default: 95%

25 "Confidence Limit" defines the probability limit for the doubt class.

- "Outlier Limit"

  Range [>0.?]

  Default 5.0

  Outlier

5   2.5.7.2 Back Error Propagation Parameters

Refer to Figure 89.

- "Learn Rate"

  Range: [0.0 1.0]

  Default: [0.1]

10   "Learn Rate" tells the network how much its weights can be adjusted

at each learning step. Setting this value high will make the network

learn fast initially, but the risk is to end up quickly in a not very good

local minimum. If the network is computing with a smaller learning

rate, it has a better change of finding a good minimum as the error

15   surface (on which we are doing a gradient descent) can be more

accurately explored with smaller steps.

- "Momentum"

  Range: [0.0 1.0]

  Default: 0.9

20   Since standard Back-Error Propagation is very slow, most

implementations add a "Momentum" term to the learning rule. This is

just a value in the interval [0,1] that is used when a new weight

change is computed. The "Momentum" value simply adds an among

of the last weight change to the current, those letting earlier weight

25   changes influence the current. The result is a tremendously improved

learning speed; however, sometimes at a minimal cost of accuracy.

Setting the "Momentum" term to high will often result in weight

oscillations where the network will not be able to start learning anything at all.

- "Epochs"
  Range: [1, 100]

5   Default: 100

"Epochs" controls how many times the whole set of input vectors are processed by the network. This default setting (and max-value) is probably too small. Ideally, the number of epochs should be decided by the system. See the next parameter.

10   - "Error limit"
  Range: [0.00, 1.00]

Default: 0.01

"Error limit" controls when to stop training. The value signifies the total RMS-error in the network output.

15   2.5.7.3 Competitive Learning Parameters

Refer to Figure 90.

- "Number of Iterations"
  Range: [1, 100]
  Default: 25

20   "Number of Iterations controls how many times the input vectors are iterated through in order to find stable cluster vectors.

- "Learn Rate"
  Range: [0.01, 0.99]
  Default: [0.20]

25   This parameter adjusts how fast the competitive network is allowed to adjust its weights in order to map the input vectors into the pre-determined number of classes. Setting "Learn Rate" high will

130

introduce more "freedom" in the clustering process, but at the cost of stability.

- "Conscience"
  Range: [0.0 50.0]

5   Default: 1.0

In the standard competitive learning algorithm, there is no guarantee that all the output classes will take a fair share of the input vectors. If we are unlucky, one output class could, in theory, win all input vectors. To guard against this, the "Conscience" mechanism ensures

10   that output classes that are winning a lot, get a "bad consciousness" and temporarily withdraw from the competition. The result is an even spread of the input vectors over the output classes. A very high "Conscience" value will slow down the clustering process.

2.5.7.4 K-Means Clustering Parameters

15   Refer to Figure 91.

- "Error Limit"
  Range: [0.0.0.1]
  Default: 0.0.1

"Error Limit" controls when to stop learning. The training is stopped

20   when the cluster vectors for the pre-determined number of clusters change less than the error limit. Low error limit values will find adjust the cluster vectors at the cost of longer training time.

2.5.8 Save Results Selection Dialog

Some classification methods provide auxiliary output data, such as

25   quality indicators in addition to the main output, the classification

result. The auxiliary output is optional. The Save Results Selection
dialog allows the selection/deselection of the output data.

Refer to Figure 92.

<u>Entry Points</u>

5   The dialog is made available for the Administration Window menu-bar
"Functions" and the convenience "Save..." in the "Active Run" area. The
button sensitivity depends upon the existence of results, i.e. a
classification must have been executed before ("Classify" button).

<u>Functions</u>

10   The dialog is exclusively valid for the current classification run in
context. The classification context is show in a read-only area on top of
the selection list.

The list "Available Data" offers all data items that are created during the
classification run to be selected for storage. The list has extended
15   selection behavior. The class-grid, as the primary output, is selected by
default. If auxiliary representations exist, they are by default selected for
storage. The convenience functions "Clear" and "Select All" allow global
list manipulations.

Since the list selection offers some pitfalls, the user might run into
20   confirmation dialogs which pop up under the following conditions after
having pressed "OK":
   • "No Output selected – Only input bindings and parameter settings
     are saved as permanent database entities. Confirm: [Yes][No]."
   • "The classification result has not been selected for storage.
25   Continue with storage of auxiliary results only? [Yes][No]"

132

At this stage of the process, after having performed a classification, it is convenient to document the process, its input binding and parameters since all components are consistent at that time. The option "Generate a Report", which is set by default, offers report generation. See "Report

5   (2.5.10)" for further details.

As already discussed in section "Data Management," certain rules for keeping consistency have to be enforced. After having left the "Save Results Selection" with "OK" and classification results selected for storage, the current classification run becomes 'read-only'. All function

10   buttons that allow input binding and parameter modifications become insensitive thus invalidating the classification run consistency

Refer to Figure 93.

The selected temporary data are saved to the database on "OK"; the read-only status is marked by setting the dialog component sensitivities.

15   "Cancel" closes the dialog and keeps the classification run status unchanged. "Help" offers help in context.

2.5.9 Class Library

The Class Library Dialog is the user interface for manipulation of the class definition repository. The dialog itself is independent of any

20   classification run. Changes in the class definition apply to any classification runs and results that refer to the class definitions having been modified.

Entry Points

The dialog is made available from the Administration Window menu-bar

25   "Options" and the "Classes in Use Dialogues/Class Properties..."

Refer to Figure 94.

The top frame "Class Hierarchy" serves as a navigation and selection device and uses the ListTree widget ["ListTree"]. The widget's (single) selection feature is used to identify a class or a class family. The class-
5   families may be packed/unpacked by selecting the folder icons; individual classes or class families are selected by clicking on their name.

The class selected and highlighted in the "Class Hierarchy" is the class in context and can be edited in the "Class Properties" frame. In the first
10   implementation phase only the "Name" and class color are editable. New attributes will be added on demand when the SeisClass class concept is generalized for use by other similar applications, i.e. RockClass. This is not expected to happened before GeoFrame 4.0.

A class can be renamed simply by entering a new name in the editable
15   text field right of the "Class Name" label. The only constraint is that the class name is unique. Uniqueness is enforced upon "OK".

New, user-defined classes can be created by selection of the "Add a Classes..." option. The class or class family currently selected is taken to derive a default for the new class' parent:
20   • if the class currently selected has children (i.e. it is a class family), then take the class family as parent
   • if the class currently selected has no children, take the class' parent as parent
   • if the class currently selected belongs to the "Built-in Classes" family,
25   no parent is pre-selected.

Refer to Figure 95.

The "Class Name" is user-defined and can directly be typed into the editable text field. It is required that class names are unique. Uniqueness is checked upon "OK".

The default parent can be altered by selecting "Parent..." and picking
5   another parent class or family via the "Parent Selection" dialog.

The same "Parent Selection" dialog appears on selection of "Reparent..." in the Class Library dialog. Here the parent-child relation of existing classes can be modified.

Color Definition

10   Class Colors are selected via the Charisma 4.0 color selector. The current settings are presented as colored buttons and an X-color name reference.

Dialog Action Buttons

The dialog operates on a 'current context' given by the selected class or
15   class family. Each modification must be committed individually, i.e. by pressing "Apply". When having modified a class' properties and changing the class context without having applied the changes, the following message will be displayed:

       "The modifications have not been saved yet! Apply changes now?
20   OK/Cancel"

On selection of "OK" in this message box, the modifications are applied before setting a new context or leaving the dialog; "Cancel" will not apply the changes.

In detail the action buttons cause the following:

135

- "OK" — Apply (see below) any modifications and remove the class library dialog from the screen.

- "Apply" — updates the class definition catalog with the modifications and notifies the application about the changes. The notification will only take place when either the name or the color of previously existing classes are changed.

- "Reset" — restores the current class' status as before modification.

- "Cancel" — does not apply any changes and removes the class library dialog from the screen.

10   • "Help" — provides the corresponding help in context.

2.6 Crossplot

The SeisClass Crossplot acts as the main inspection and selection device for data in attribute space – view many or all attributes of one position simultaneously. The IMain Seis and IMain Map windows serve the opposite purpose – one attribute at all positions.

Since there is no limitation on the number of attributes, the Crossplot display is able to handle any number of dimensions in attribute space. The following figure shows the 2D case as a simple, comprehensible example and a 4D case representative for N-dimensional attribute spaces.

Attribute spaces having more than two dimensions will be mapped into the two-dimensional display space (Crossplot) by a projection function.

The parameters of the projects functions can be changed interactively to visualize a translation and rotation of the Crossplot projection place.

A "Histogram" window is optional available. It appears below the multi-frame crossplot area. Data displays do always refer to the currently

136

active frame and its data visibility selection. Histograms of individual attributes or projection on to crossplot frame axes can be evaluated.

New to pure visualization, the crossplot window serves as data query and selection tool. Point properties and its origin are monitored in the
5   information area. Various selection methods are offered to defined two-dimensionally shaped areas on crossplot. Two-dimensional selection areas intersected with a N-dimensional volume correspond to a N-dimensional sub-volume that determines whether a point is selected, i.e. inside the sub-volume, or not.

10   The selection or display manipulation function currently active in the crossplot frame is also active for the histogram frame. Crossplot's arbitrary area selection gets replaced by a bin selection for the histogram. This allows the identification of outliers in the distribution, either of selected data or of training/validation data.

15   The SeisClass Crossplot window is a new graphic window, parallel to the existing IMain graphic windows. (See "SeisClass and IMain Windows".) The window layout will be similar to the InDepth crossplot, having a "Histogram" on the left, one "active Crossplot Frame" and up to eight "passive Crossplot Frames" as main components and a standard
20   "Histogram" at the bottom. The following figure presents the SeisClass crossplot window layout.

Refer to Figure 96.

Two main categories of data can be visualized in crossplot frames, either separate or combined:
25   • "Current Selection"
       Attribute vectors from selected areas on the map window (horizontal classification domain) or the seismic section window (vertical classification domain).

137

- "Training/Validation"

  Attribute vectors from the training/validation areas defined in the input binding of supervised classification runs.

These categories are exclusively based on the current active classification run ("Active Run"). It provides the input binding, i.e. the link to the attributes and – in case of supervised methods – training/validation data selections. Since the training/validation set is structured, sub-selections are offered.

Attribute vectors are shown in crossplot frames as scattered points, color coded by their associated class or 'current selection' color. The points symbols correspond to the point type, their size is a display option.

Refer to Figure 97.

Based on the specific selection context, a variety of viewport setups can be defined via the "Crossplot Functions (2.6.1.1)" pull-down menu. Once a coordinate system has been established, it can interactively be modified through zoom, pan or rotation.

2.6.1 Crossplot Menu-bar

The "File" pull-down contains only one button, allowing the termination of a SeisClass session. The button label will read "Exit" if SeisClass has been launched as a separate application or "Close SeisClass" if SeisClass has been started from an IMain session.

2.6.1.1 Crossplot Functions

Refer to Figure 98.

The first group of functions deals with interactive viewport manipulations.

"Zoom" changes the active frame's scaling factor. "Zoom" is an interaction on the currently active frame. This is accomplished by

5   putting up a rectangular area that is going to be the future frame boundary. The "Zoom" function stays active until it is replaced by any other interaction. The zoom window is marked by a rubber-band rectangle during dragging (MB1 pressed). Zooming is executed on MB1 release. While active, the One Line Help provides the message "MB1:

10   Drag a rectangle to zoom in; MB2: reset zoom; MB3 zoom out 151%" when moving the cursor on the active frame.

The "Reset Zoom" function is selection sensitive. Similar to reset functions on the map or section window, the extent of the selected data in the current projection is the basis for a scaling factor. Since the

15   number of selected data points and the projection may vary drastically from time to time, Crossplot' Reset Zoom" adjusts the scaling factor so that all points are visible in the viewport. Note that "Reset Zoom" does not change the current interaction context.

"Rotate" activates an interaction to change the current frame's

20   coordinate system projection. A spheroid surface fit to the current frame is dragged by the cursor.

Refer to Figure 99.

The Rotate function stays active as long as it is replaced by any other interaction. While active, the One Line Help provides the message "Drag

25   the Ellipsoid lying in the frame; MB2: Quit Rotate" when moving the cursor on the active frame. The rotation is performed in real-time: a rubber-band vector from the cursor position to the viewport center

symbolizes the rotation handle. Cursor tracking is disabled during rotation (MB1 pressed) and re-enabled on MB1 release.

"Pan is a convenience function to perform translations of the viewport center. The translation vector is defined by the start and end point of a

5   cursor drag interaction. The vector is presented as rubber-band during the drag operation. "Pan" is executed on MB2 release. While active, the One Line Help proves the message MB1 Drag to Pan; MB2: Quit Pan" when moving the cursor on the active frame.

The second group of functions deals with editing selection and

10 training/validation data. A sub-dialog (see "Training/Validation Edit [2.6.2]") controlling all selection and editing activities is available via "Edit Training/Validation..."; most interaction functions are available as icons or menu functions, so that expert users can leave as much screen space as possible for crossplot itself.

15 The "Graphic Selection" mode can be set in a cascading toggle menu. See also "Section 2.6.2" and Table 3, "Selection Function Button Bindings".

The definition of training/validation data requires a "Class in Context..." selection which is performed in a small list selection dialog.

"Data Type" cascades to a toggle menu to a set "Training" or "Validation"

20 as data type in context. Details are discussed in "Section 2.6.2".

"Add to Class" and "Remove from Class" perform the final editing actions. Both functions refer to a current selection which is either added or deleted from the current training/validation set (classification run context; see "Training Validation Set Dialogue [2.5.5]". These functions

25 usually invalidate an intermediate classification result since it is a modification of the input data.

140

The last group of functions covers coordinate system setup and projection manipulations in general.

"Reset Projection" resets the current frame's coordinate system according to the settings defined in the "Data Visibility (2.6.9)" dialog. The selected
5   attributes are represented by their unit vectors. The attributes are used directly in the 2D case. When three or more attributes are selected, an averaged vector is derived from the selected attribute unit vectors. This average vector defines the out-of-screen direction, the first attribute in the list ("Data Visibility") point in the vertical direction. A reset zoom is
10   implicitly activated in order to restore a meaningful display.

Refer to Figure 100.

"Principal Component Projection" offers a data dependent coordinate system. The data selected in the "Data Visibility(2.6.9)" dialog are subject to a principal component analysis. The eigenvectors corresponding to the three largest eigenvalues build the new coordinate
15   system. The initial display will follow the same procedure as shown in "Figure 62" – "A" being the main principal component axis, "B" and "C" the small principal component axes.

"Multiple Magic Perspective" is only sensitive in a multi-frame situation.
20   This function takes the current frame's coordinate system and varies the perspective systematically. The non-active frames are updated each having a different perspective.

2.6.2 Training/Validation Edit

The Training/Validation dialog collects all functional controls and
25   context selections that are required for editing Training/Validation data in crossplot frames. Each element of the dialog is also available from the icon column and the "Crossplot Functions" menu-bar pull-down. This

dialog addresses specifically the untrained user who is not yet familiar with SeisClass functions.

Except for the "Data Selection", all other dialog components are only sensitive (and meaningful)in a supervised classification method context.

5   Refer to Figure 101.

The dialog and its functions refer to the point cloud in the current active crossplot frame. Data selection interactions are initiated from this dialog that lead to a graphic selection, i.e. a marked point cloud. This selection can be associated to a class and added to or removed from the

10 training/validation set in context. (See section "Training Validation Set Dialogue 2.5.5".)

Entry Points

The Training/Validation dialog is only available via "Crossplot Functions" menu-bar pull-down.

15 Functions

The upper frame, "Data Selection", offers the data selection modes available on the active crossplot frame. Simultaneously, it shows the current active selection mode. The selection modes are interactions on the current active crossplot frame. They are active until they are

20 replaced by any other interaction, like "Zoom" or "Rotation". The toggle buttons show whether and which selection mode is active. The push buttons right of the toggles activate the respective crossplot interaction – like the toggles – but are regarded as more intuitive function activators than the mode selectors.

142

The following "Table 3" specifies the graphic selection behavior and "Table 4" the 'One Line help' texts.

Table 3: Selection Function Button Bindings

Selection Function	Action	MB1 left Mouse Button	MB2 middle Mouse Button	MB3 right Mouse Button
	Press			
Single Point	Drag			
	Release	Select nearest Point		Deselect nearest Point
	Press	Select Points inside 'Brush'a)	Start 'Brush'a) size definition	Deselect points inside 'Brush'a)
Paint Brush	Drag	Select Points inside 'Brush'a)	Change 'Brush'a) size	Deselect points inside 'Brush'a)
	Release		Start 'Brush'a) size	
	Press	Start/continue digitization		
Polygon	Drag	Add polygon points continuously b)		
	Release	Add a polygon point	Select all points inside polygon	Deselect all points inside polygon

5 a) 'Brush' is to be understood as a rectangular selection area.
  b) The motion sensitivity is set to delta X/Delta Y>=3 Pixel.

Repeated activation of the selection functions add data to or remove data from the current selection. The selection may be supplemented by selection done in other windows. Each data point can only be selected once. The selection buffer manipulation function takes care of this.

5

Table 4: One-Line Help for SeisClass Selections

Selection Function	Message
Single Point	MB1: Select nearest point; MB3: Deselect nearest point
Paint Brush	MB1: Brush select points; MB2: Drag brush size; MB3: Deselect points
Polygon	MB1: Digitize polygon; MB2 Close and Select; MB3 Close and deselect

A selection is cleared by activating the "Clear" button in the "Data Selection" frame.

The next block "Available Classes" presents a list of classes in use in
10   conjunction with the current classification run context (the list is synchronous with the definitions done in "Classes in Use Dialogues (2.5.6). A selection of a class in context is mandatory. If no class is selected, the "Add" and "Remove" buttons stay insensitive.

Selections must be tagged either as Training or as Validation data when
15   they are added or removed from the set. The data type in context is changed for the entire crossplot window. The context indicators are updated as well as the equivalent toggle buttons in the "Crossplot Functions" menu-bar.

Data that are highlighted as current selection can be added or removed from the current training/validation set by pressing the "Add" or "Remove" button. The action is applied instantaneously.

The status of a supervision sample is either "Training" or "Validation".

5   Note that adding a selected sample to the training set which was previously a validation set member implies a removal of the sample from the validation set.

The current selection is kept active, independent of any assignments by "Add" or "Remove" and can be supplemented or refined. Alternating

10   "Add" selection "Remove" selection causes an undo or redo selection.

Since the dialog controls mostly modes and contexts and "Add/Remove" actions are executed immediately, only a Close button is required to remove the dialog from the screen.

Refer to Figure 102.

15   2.6.3 Multiframe

Crossplot Multiframe appears similar to the familiar Charisma Multiframe dialog. It has, however, been enhanced for the specific SeisClass crossplot purposes when creating new frames.

Entry Points

20   The Multiframe dialog is only available via "Crossplot Options" menu-bar pull-down.

145

Functions

As for the other Charisma windows, up to nine frames may be opened at the same time. The geometry can be set by pressing the respective

5 "Multiframe" button; it will activate all neighboring frames in upper left direction. The buttons in the Multiframe matrix show the status via color.

The frames may have individual scroll bars. This option can be toggled; the default is common set of scroll bars since a pan function exists and

10 this option leaves more space for the individual frames.

Rules on how to fill newly created frames initially can be set via the "New Frames" option menu. The frames are either left empty, filled with a copy of the current active frame or filled with the result of a systematic variation of the current active frame's projection perspective (Magi

15 Perspective). This function is identical to the "Crossplot Functions" entry.

2.6.4 Crossplot Display

Entry Points

The crossplot Display Options dialog is only available via "Crossplot Options" menu-bar pull-down.

20 Functions

This dialog collects all crossplot display option selections, namely annotation, point symbol size and display priority.

Refer to Figure 103.

The "Crossplot Annotation" frame controls via toggle buttons:

- "Show Axes" – the attribute axes are shown as vectors (default on) pointing away from the center of the viewport,

- "Attribute Names " – the axes are annotated by their attribute names (default on),

- "Well Names" – whether attribute vectors derived from well logs should be annotated by the well name (default off, insensitive if no well data are available).

In the "Symbols" section, predefined symbol sizes can be associated with the data classes that can be displayed in crossplot frames. The symbol size can be used to emphasize certain data types relative to others. The symbols are offered in real size pixmap option buttons.

The "Display Priority" section finally allows the specification of a display sequence after data types and classes. Together with the symbol size, these options offer full flexibility to visualize and highlight specific classes or data types with respect to others.

The contents of the list reflect the settings in the "Data Visibility" dialog. The list operates in single selection mode. The selected list item can be moved upwards or downwards by one position when clicking on the respective arrow buttons on the right.

The convenience toggle button "Apply to all Frames" propagates the selections in this dialog to all other crossplot frames if it is set and "OK" is activated. Default setting is 'apply to current frame only'.

Since this dialog performs only non-destructive actions and selections, only "OK", "Cancel", and "Help" action buttons are provided.

2.6.5 <u>Histogram Display</u>

The Histogram Display dialog provides the necessary tools to control the data presentation in the crossplot histogram window. Since the histogram window inherits the data visibility and display settings from the current active crossplot frame, they are also implicitly valid for the

5 histogram. (See "Crossplot Display" and "Data Visibility.") General, the current active selection or display manipulation interaction is also active for the histogram.

The specification cannot cover the histogram options in the required detail since the commercial widget package that is going to be used for

10 the implementation is not yet accessible. Only basic features, independent of the widget type being used, are presented in this section. The detailed display manipulation options remain to be specified.

Entry Points

The Histogram Display dialog is only available via "Crossplot Options"

15 menu-bar pull-down.

Refer to Figure 104.

Functions

The widget independent options seen so far are:
- Presentation

20   toggles whether the frequency bars are drawn as outline or as filled rectangle; the color always corresponds to the class or selection color.
- Scaling
       toggles whether the vertical axis (frequency) is normalized individually ("Individual") or scaled to the maximum frequency of all

25   displayed histograms ("Common").

A control for the specification of the histogram bin size might be required. An automatic bin size calculation is being considered, following the empiric rule:

N of bins = C_1 + C_2 . lg(N number of samples) where C_1 = 4, C_2 = 3/2.

5 **(Eq. 1)**

Since this dialog is going to be substantially extended, a full set of action buttons are proposed.

2.6.6 Information Area

This dialog follows the style of the common window's information area
10 customization dialogs. (See "Histogram".) It contains, however, only crossplot relevant items. The information items on display refer to a crossplot data point next to the cursor position. No information will be displayed if the cursor to point distance exceeds three (3) screen pixels. The information area update is controlled by the settings done in
15 "Cursor Tracking".

Refer to Figure 105.

Entry Points

The Information Area customization dialog is only available via "Crossplot Options" menu-bar pull-down.

20 ### Functions

The user interactions and dialog handling is identical to the other information area customization dialogs. The list of available informational items is described in the following table:

149

Tabl 5: Crossplot Information Area It ms

Item	Description
Attribute Vector (A1 A2...AN)	Lists the attribute vector components; the values are followed by their [unit]; the value sequence corresponds to the attribute sequence in the "Data Visibility" sector
Class	shows the class name
Data shown/Not Shown	relates the number of visible points in the active frame to all data points set in "Data Visibility".
Data Source	Either "Grid", "Section", or "Well"
Geographical Coordinates (Lat Long)	Current point's position in Latitude/Longitude
Histogram Bin	Bin interval boundary values, i.e. [V1 V2]
Projection Coordinates (X Y)	Current point's position in rectangular coordinate (project coordinate system)
Seismic Position	'Current point's position in Inline/Crossline index number
Structure Reference Value	Value of the structural reference grid at the 'current point's position (depth or time)'
Supervision Type	Either "Selection", "Training", or "Validation"
Zoom Factor	Scaling factor: 1.0 corresponds to a viewport scaling in "Reset Zoom" condition, i.e. all points in the selection are visible.

"OK" and "Apply" initiate the information fields in the information area and register them for being updated by "Cursor Tracking". "OK" and

150

"Cancel" remove the dialog from the screen. "Reset" restores the dialog settings according to 'last applied'. "Help" provides help in context.

2.6.7 Cursor Tracking

Refer to Figure 106.

5   The dialog's layout and human interface is identical to the common cursor tracking dialogs on the other windows. However, the event handling is adapted to the purposes of the crossplot window.

Send

Since a continuous cursor position monitoring in a general N-
10 dimensional attribute space is impossible, the cursor is snapped to the nearest data point if the screen projection distance between cursor and data sample is less than or equal to three (3) pixels. The point is identified in the internal display buffer and traced back to a position. This position information is broadcast to other registered windows as
15 "Send" is activated.

Tracking events are not generated during all interactive processes where cursor dragging is involved like rotation, panning, zoom window definitions, paintbrush selections and selection polygon definitions.

Receive

20 Crossplot cursor tracking receives position information on the cursor and will build up an attribute vector corresponding to this position. This attribute vector is displayed as track-cursor in the crossplot.

2.6.8 Tour Options

151

Refer to Figure 107.

The Tour Options dialog offers a slider selection for the Rotation
Increment used for the Start Tour function.

2.6.9 Data Visibility

5   This dialog allows the selection of the attributes defining the coordinate
system and a data visibility selection according to their category.
Selections can be applied to the active frame which is the default or to
all frames. Data visibility is expected to be changed frequently; the Apply
button activates the Selection keeping the dialog for convenience. The
10   dialog is sensitive for changes in the current classification context,
notably for changing the input attribute set.

Refer to Figure 108.

Entry Points

Data Visibility is only available via "Crossplot Selections" menu-bar pull-
15   down.

Functions

Attribute visibility is controlled via a list in extended selection mode. The
selection of two attributes is a minimum requirement; it is enforced on
Apply. The list presents all the attributes by measurement which are
20   associated with the input attribute set in context for the current
classification run. The grid identification is given as an addition
navigation aid since it is expected that some attributes may have
identical measurements, i.e. "Amplitude" or "Dimensionless".

Two main data categories can be visualized in crossplot, a "Current Selection", i.e. data without a class coupling, and class data (controlled under "Class Visibility"), e.g. results and/or "Training/Validation" data. Both categories refer to the attribute selection above. It should be noted that the options under "Training/Validation" are only sensitive in the case of supervised classification methods. The category "Current Selection" is unstructured, whereas the result selections and training/validation categories offer sub-selections. One may want to focus on certain classes or separate the data according to their source or supervision type.

The "Current Selection" data category can be toggled on or off independently. Classes are chosen from a list in extended selection mode. The classes offered correspond to the classification run context. See "Classes in Use Dialogues (2.5.6)." Class selections are valid for the "Histogram Data Visibility" dialog and communicated on "OK/Apply". Further sub-selections according to its supervision type under "Training/Validation" are performed through toggles. Alternatively or in addition, training and validation data visibility can be determined by source.

The data display is updated on "OK" and "Apply", following the toggle "Apply to all Frames" in the active or in all frames. The current frame's projection is kept as is if the attribute visibility selection stayed unchanged. Changing the attribute visibility causes a reset on projection of the current active frame, identical to the menu-bar function "Reset Projection". "Cancel" does not change any selection but removes the dialog from the screen. "Help" provides the usual help in context.

The default settings for this dialog are:
- The first (maximum three) attributes in the "Attribute Visibility" list selected,

153

- "Current Selection" enabled for display,
- all "Available Classes" selected as visible,
- all supervision and source types set visible.

Refer to Figure 109.

5   2.6.10 Histogram Data Visibility

What the "Data Visibility" is for crossplot frames is this dialog for the histogram frame. Since the histogram shows the same data as the active crossplot frame, the most important selections of the "Data Visibility" dialog are inherited.

10   The "Attribute Selection" list offers all attributes that were selected to be active for the active crossplot frame. The list is supplemented by the projection onto the active frame's X and Y axis. Only one attribute or axis can be selected.

The "Class Visibility" refers to the parallel display of histograms showing
15   the data distribution per class. Classes are selected from the list "Available Classes" in extended selection mode: the list includes only those classes that have been made visible through "Data Visibility".

The section "Training/Validation" controls some options for monitoring the data distribution of supervision data. "Individual histograms per
20   Class" will compute and display a histogram graph for each selected class when activated or one single composite graph if the option has been activated. Multiple histogram graphs can be shown as Stacked (Summed) Histogram" if activated or as multiple graphs directly overlaid.

154

The data distribution of the whole attribute can be selected for computation and display as well. It may be used as a reference distribution with respect to any sub-selection.

2.6.11 Histograms

5   The detailed specification depends upon the type of commercial widget that is going to be used with SeisClass.

It is intended to provide a selection method similar to the crossplot data selections (discussed in section "Training/Validation Edit") also for the histogram frame.

10   The histogram frame is always 'active' since it relates itself to the current active crossplot frame.

2.7 GridMap Extension

The Charisma Map window plays a very important role for classification in the horizontal domain. It serves as the primary attribute grid display visualization tool, a task which it fulfills already today. It offers a

15   selection platform for areas likely to have similar properties thus belong to a common class. Finally it is the proper place to present the result of a classification, the class map.

This chapter focuses on the functionality that is not yet available, but

20   especially required by SeisClass.

- Class Map as new data presentation method ("Class Grid Display")
- Training/Validation Data display ("Training/Validation Display on GridMap")

- selecting or editing training/validation data ("Area Selections on Map")

Since the class map is introduced as a new display object, the:

- "Scaled Hardcopy" function

5   must also be enhanced to cover class map and training/validation map displays.

Refer to Figure 110.

The concept of external map display control has already been introduced in InDepth where a dedicated display function causes a complete multi-

10   grid or multi-scattered set display in the map frames. The update logic follows InDepth.

The display of multiple attribute grids being grouped in an input attribute set or other maps relevant to a classification run can be initiated from the SeisClass administration window via a dedicated

15   "Display" function. See "Navigation, Selections and Context Activation" for further details. The current active selection in the List Tree widget will be transferred, it at all possible, to a "Multiframe" map display. This implies that both individual grid representations as well as entire attribute sets can be dispatched to be displayed on the map. A grid

20   selection from the map is still possible, but the increasing number of attribute grids will make navigation more difficult.

2.7.1 Class Grid Display

In contrast to ordinary attribute grids, class grid values are discrete values, referencing a class description. Class maps represent the classes

25   by their color code. The proportional attribute value to color mapping is replaced by a direct mapping into the discrete look-up table. While

156

conventional look-up tables are read from a color scale definition file, the discrete class color look-up table must be set up dynamically.

When displaying or updating a class grid, the class_decode relation is used, giving a convening overview of the classes in use for the given

5   representation. (See "2.3.2 Class Representations" and "Class Representations".) This class list maps dynamically into the discrete class color look-up table. The mapping of the class Shale can be followed through the example.

Refer to Figure 111.

10   The class map annotation utilizes the color bar display in conjunction with the map window's information area. When moving the cursor on a class map or on the color bar, the class name at the cursor position is monitored in the information area. "Class Name" is anew member in the map information area parameter catalog ("Table 6: Map Information Area

15   Items").

Class maps require only limited edit options. Changing class grid cells is a forgery of the classification process result. Only an 'interpretive' delete option will be provided. GridMap's "EditMap" cascade pull-down menu will reflect the restricted edit options if a class grid is displayed in the

20   active frame.

Rule 34:   If a class grid is displayed in the active GripMap frame, the
              "Edit Map" menu are set insensitive, except;
              • Define Area
              • Delete Area
25   • Delete All Inside Area
              • Delete
              • Save Edits

157

See Figure 21 for a Class Map Display example.

The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such
5   modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.